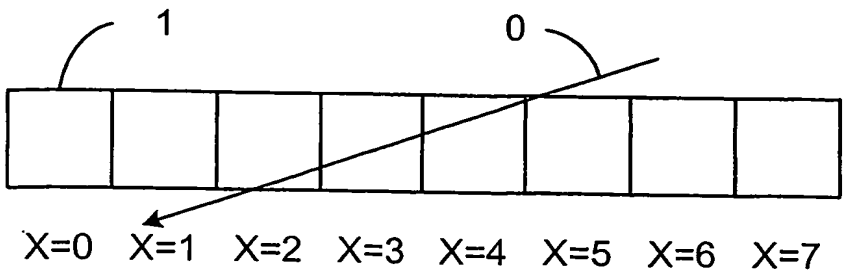


Fig. 1(a)

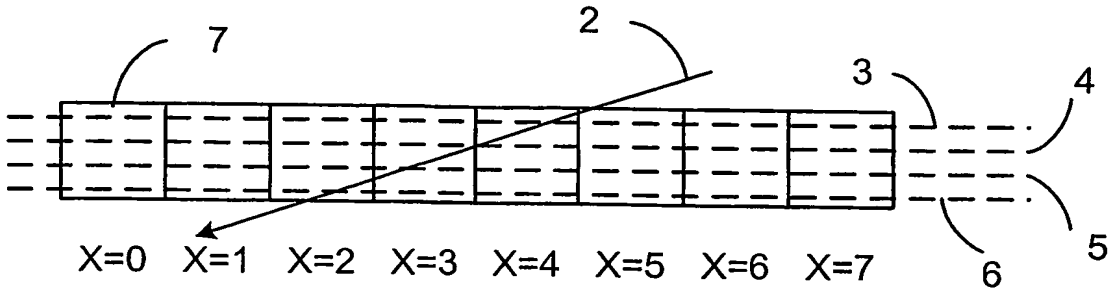


Fig. 1(b)

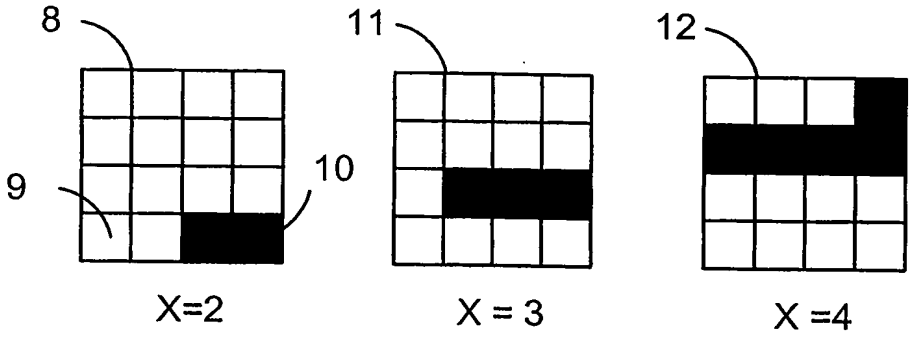


Fig. 1(c)

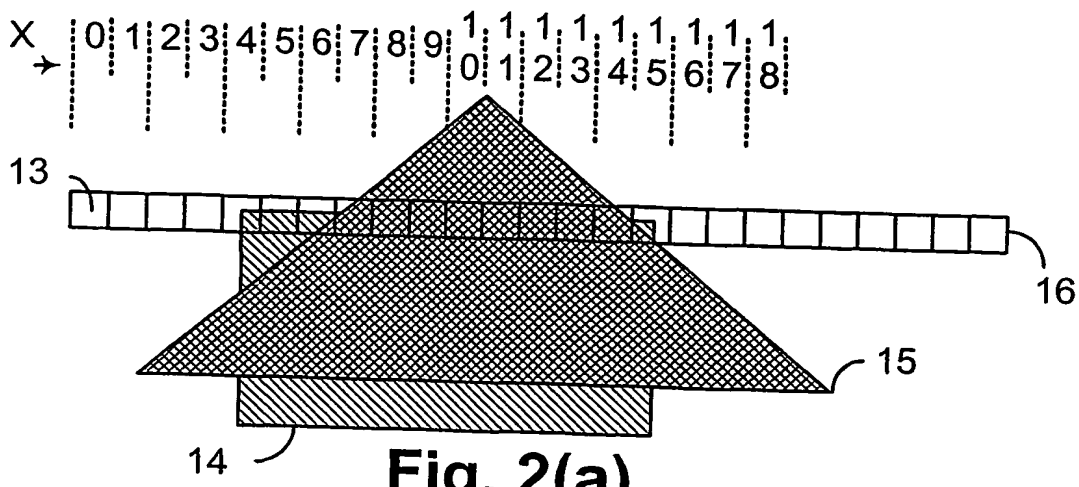


Fig. 2(a)

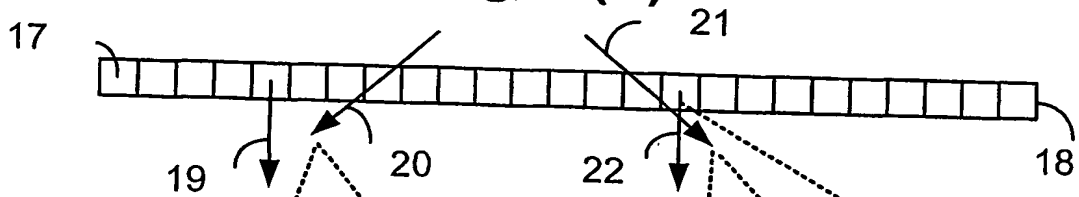


Fig. 2(b)

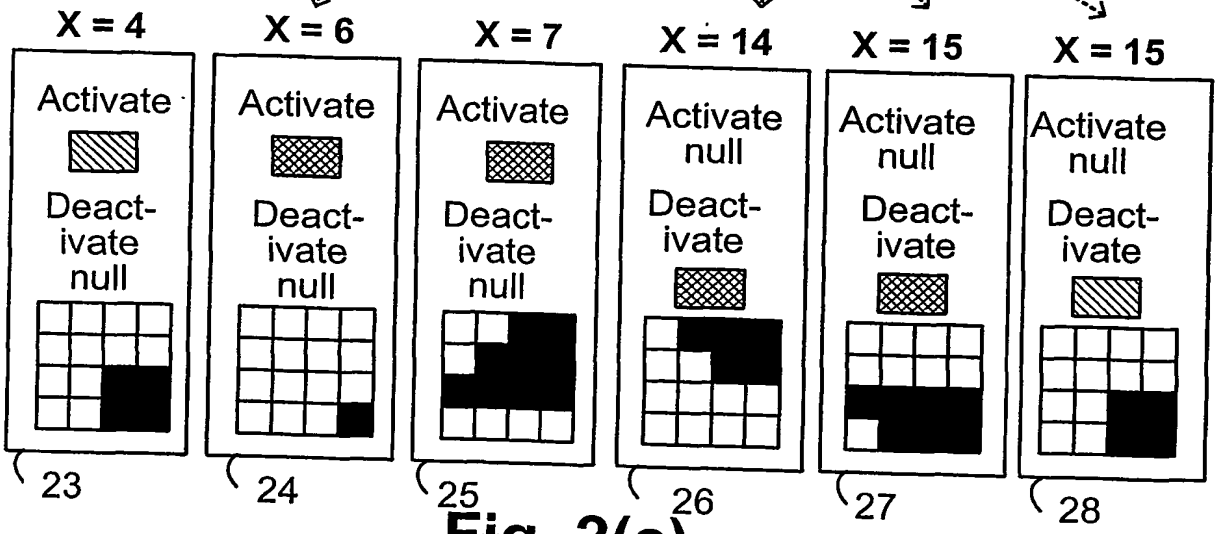


Fig. 2(c)

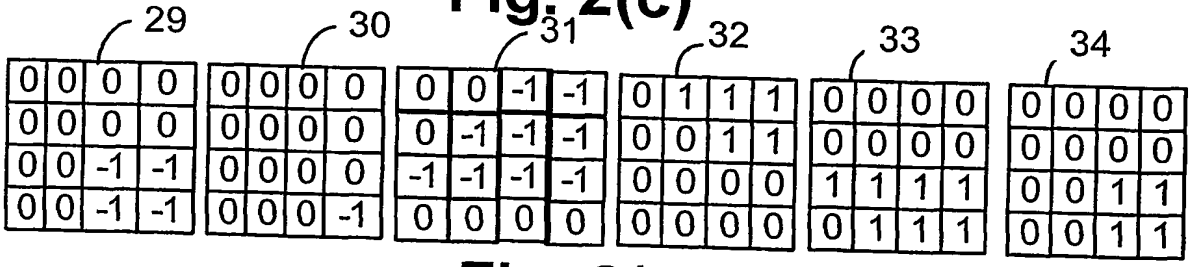


Fig. 2(d)

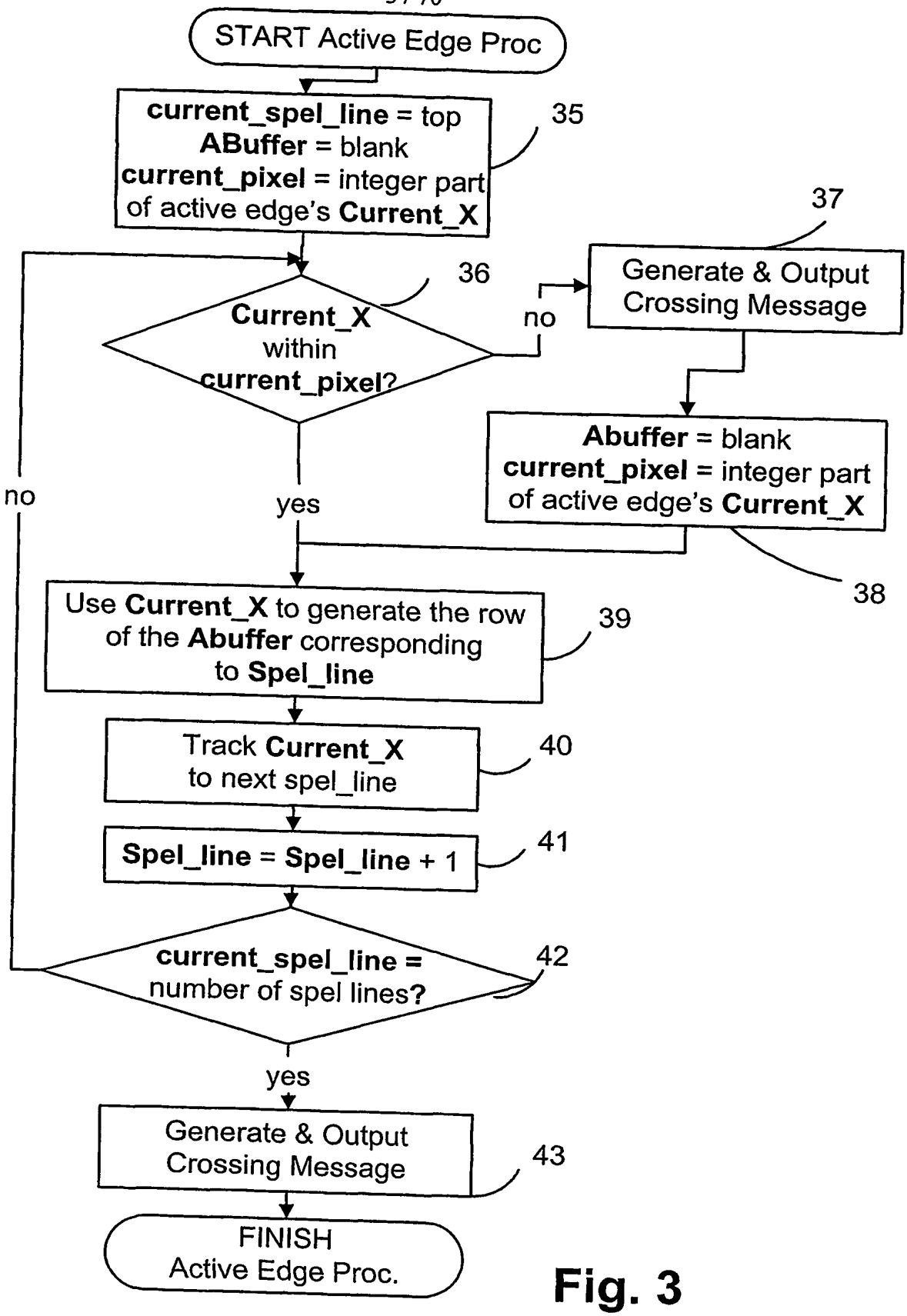

START Active Edge Proc
current_spel_line = top
ABuffer = blank
current_pixel = integer part of active edge's Current_X
35
37
36
Current_X within current_pixel?
no
Generate & Output Crossing Message
Abuffer = blank
current_pixel = integer part of active edge's Current_X
no
yes
38
Use Current_X to generate the row of the Abuffer corresponding to Spel_line
39
Track Current_X to next spel_line
40
Spel_line = Spel_line + 1
41
current_spel_line = number of spel lines?
42
yes
Generate & Output Crossing Message
43
FINISH Active Edge Proc.

Fig. 3

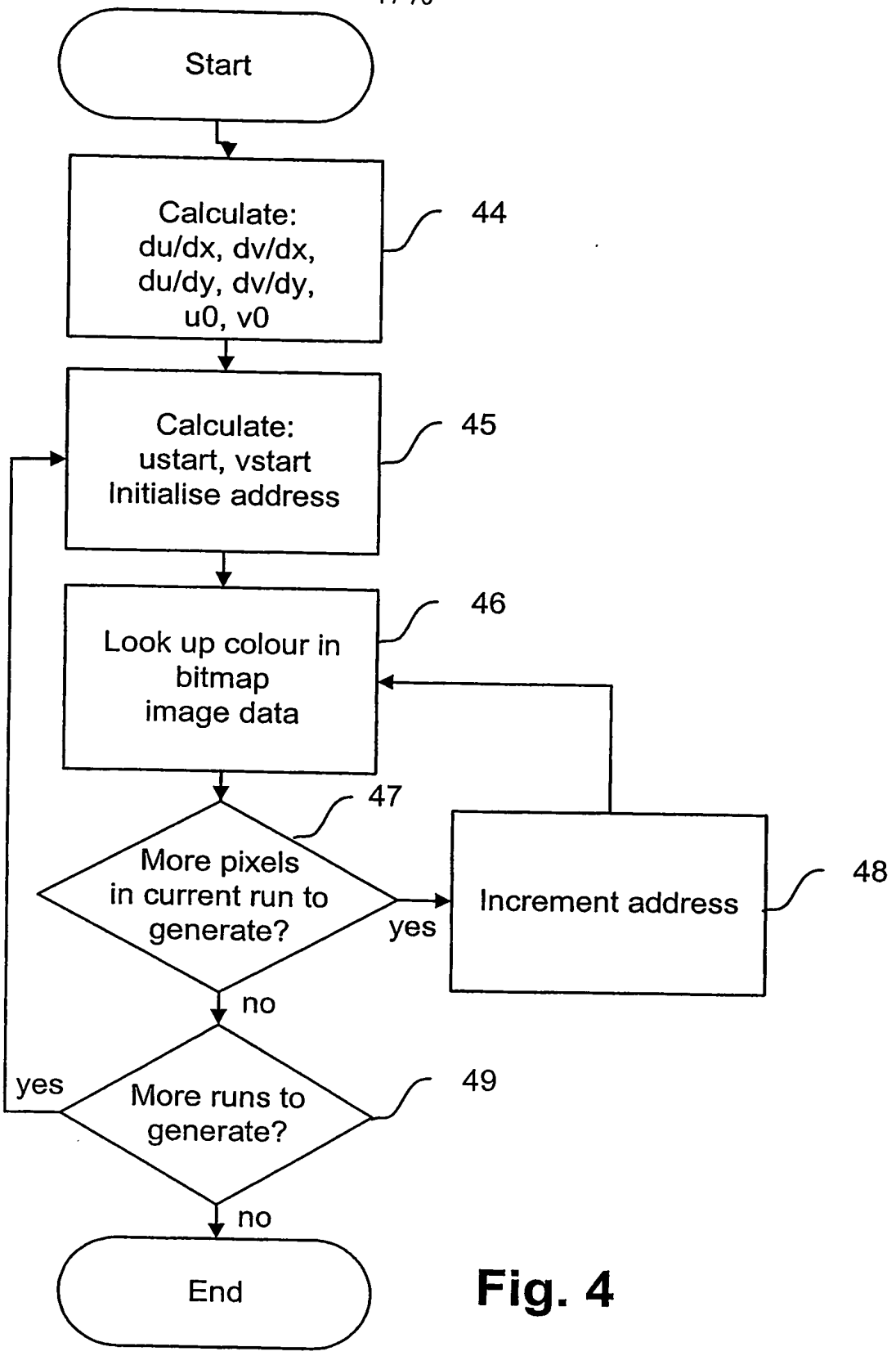

Start
Calculate: du/dx, dv/dx, du/dy, dv/dy, u0, v0
44
Calculate: ustart, vstart Initialise address
45
Look up colour in bitmap image data
46
47
More pixels in current run to generate?
yes
Increment address
48
no
yes
More runs to generate?
49
no
End

Fig. 4

```
void
DrawLine
(
    int Xs,
    int Ys,
    int Xe,
    int Ye
)
{
    int     tx = Xe - Xs;
    int     ty = Ye - Ys;
    int     grad = tx / ty;
    int     ient = tx % ty;
    int     err = (2 * ient) - ty;
    int     delta_err1 = 2 * ient;
    int     delta_err2 = 2 * (ient - ty);
    int     x = Xs;
    int     y;

    for (y = Ys; y <= Ye; y++)
    {
        SetPixel(x, y);
        if (err < 0)
        {
            err += delta_err1;
            x += grad;
        }
        else
        {
            err += delta_err2;
            x += grad + 1;
        }
    }
}
```

Fig. 5

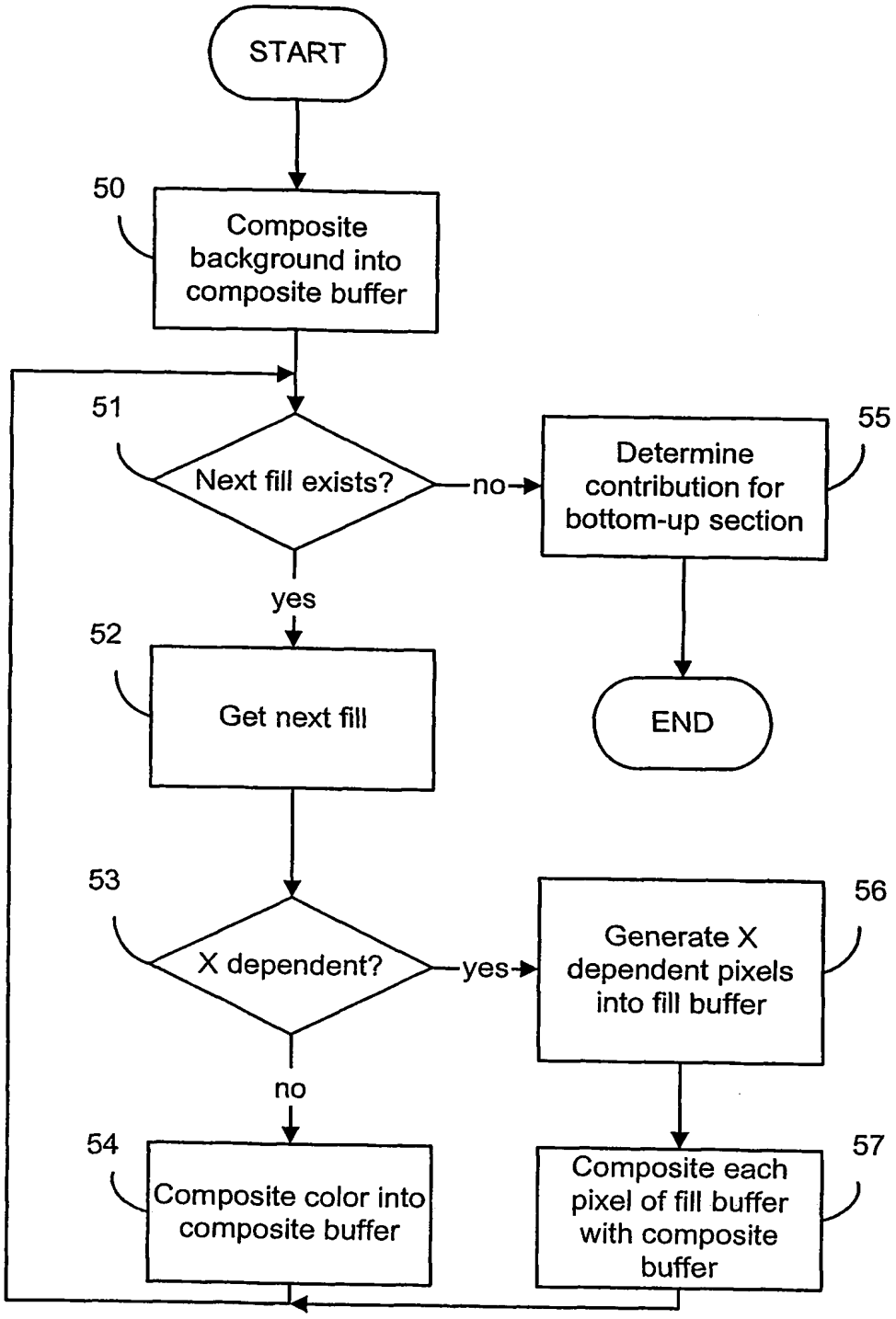

START
50
Composite background into composite buffer
51
Next fill exists?
no
55
Determine contribution for bottom-up section
yes
52
Get next fill
END
53
X dependent?
yes
56
Generate X dependent pixels into fill buffer
no
54
Composite color into composite buffer
57
Composite each pixel of fill buffer with composite buffer

Fig. 6

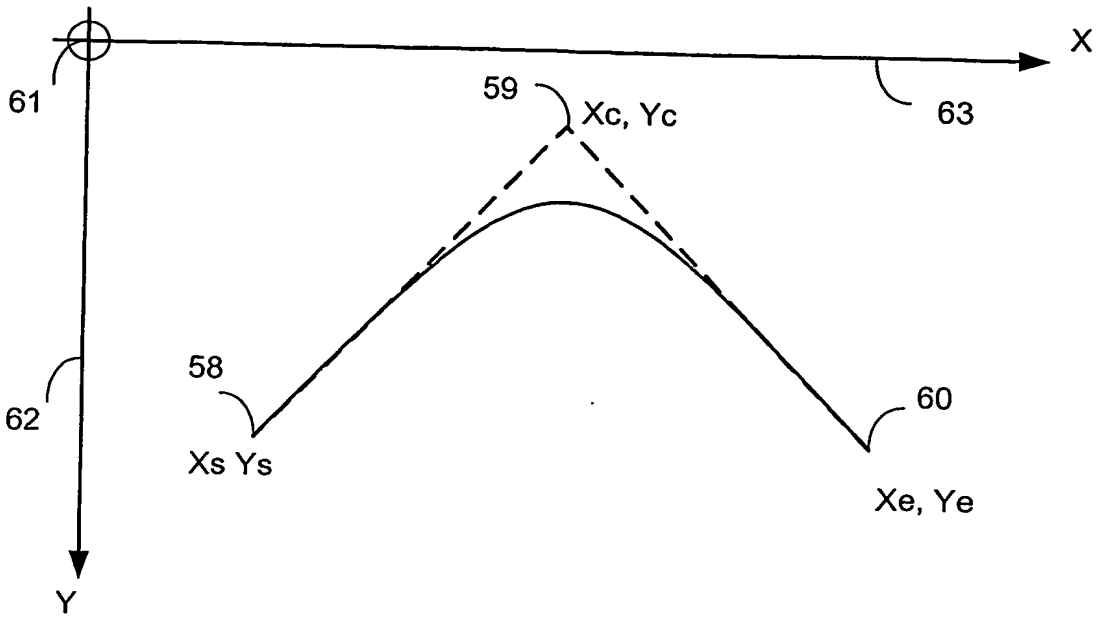

X
61
59
Xc, Yc
63
58
60
62
Xs Ys
Xe, Ye
Y

Fig. 7

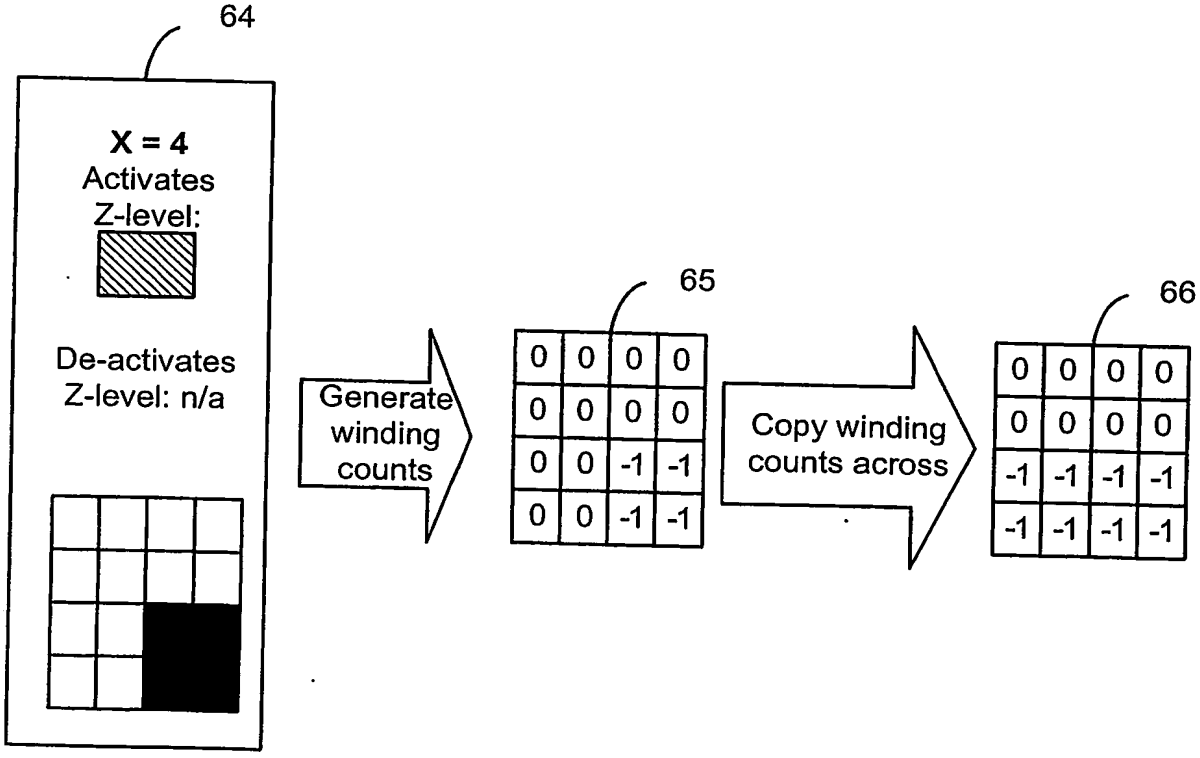

64
X = 4
Activates
Z-level:
De-activates
Z-level: n/a
Generate
winding
counts
65
0 0 0 0
0 0 0 0
0 0 -1 -1
0 0 -1 -1
Copy winding
counts across
66
0 0 0 0
0 0 0 0
-1 -1 -1 -1
-1 -1 -1 -1

Fig. 8

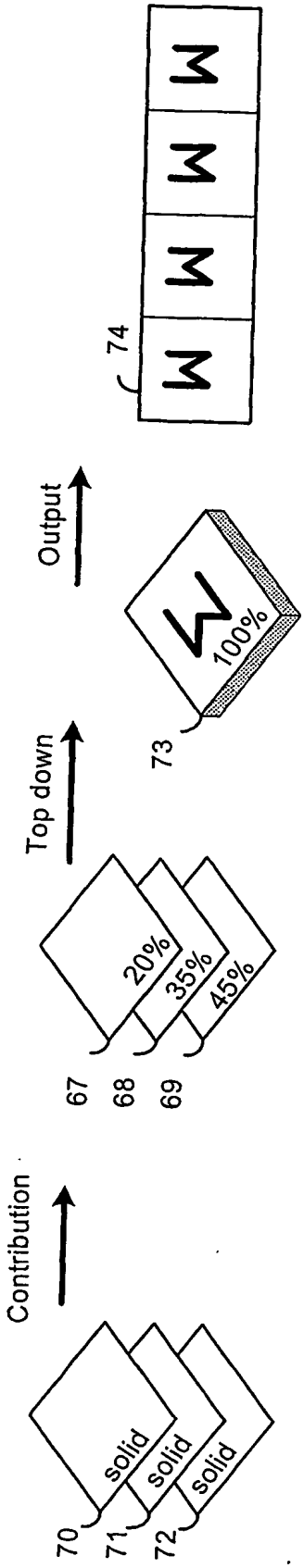

Contribution
Top down
Output
solid
solid
solid
20%
35%
45%
100%
M
M
M
M
70
71
72
67
68
69
73
74

Fig. 9(a)

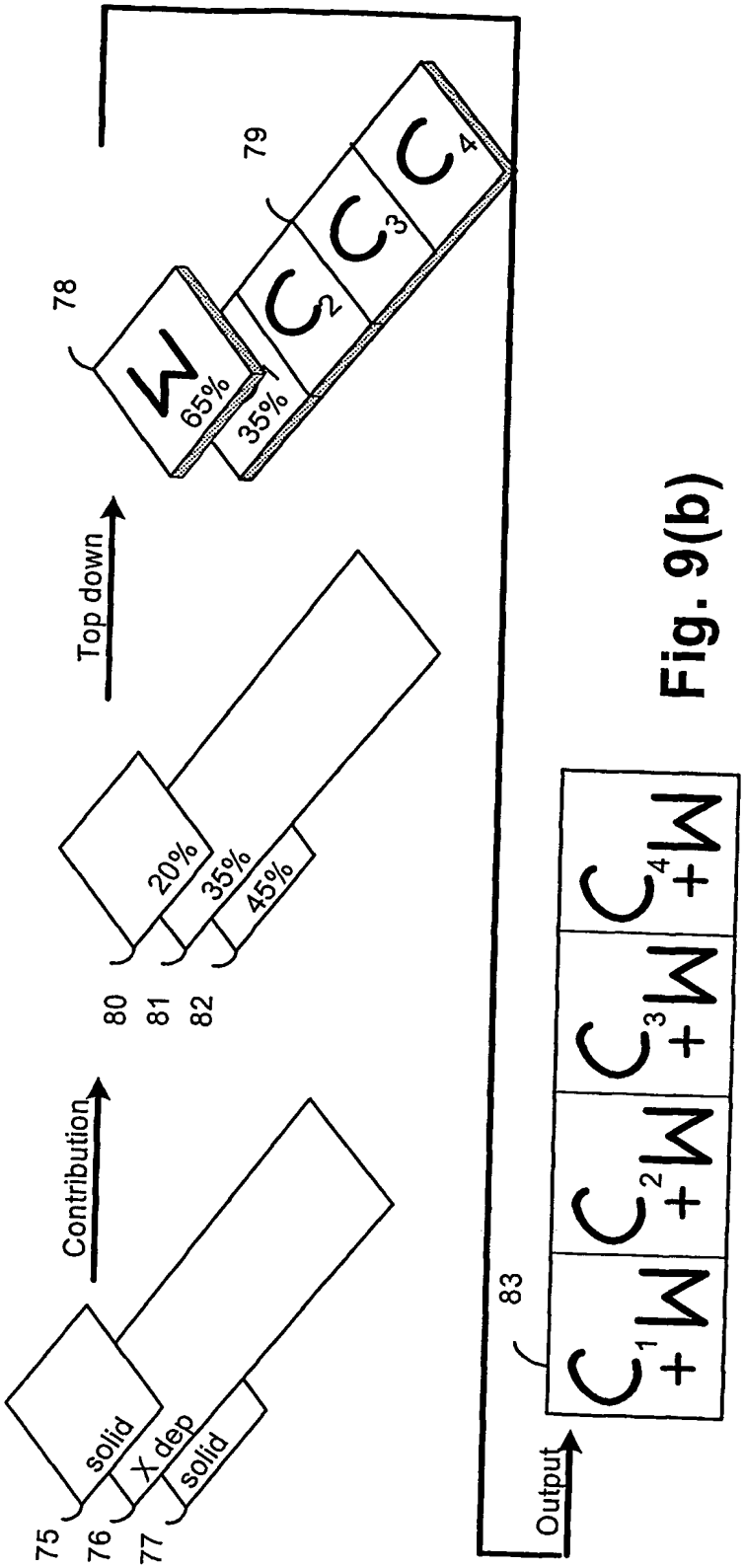

Contribution
Top down
Output
solid
X dep
solid
20%
35%
45%
65%
35%
C1 + M1
C2 + M2
C3 + M3
C4 + M4
75
76
77
80
81
82
78
79
83

Fig. 9(b)

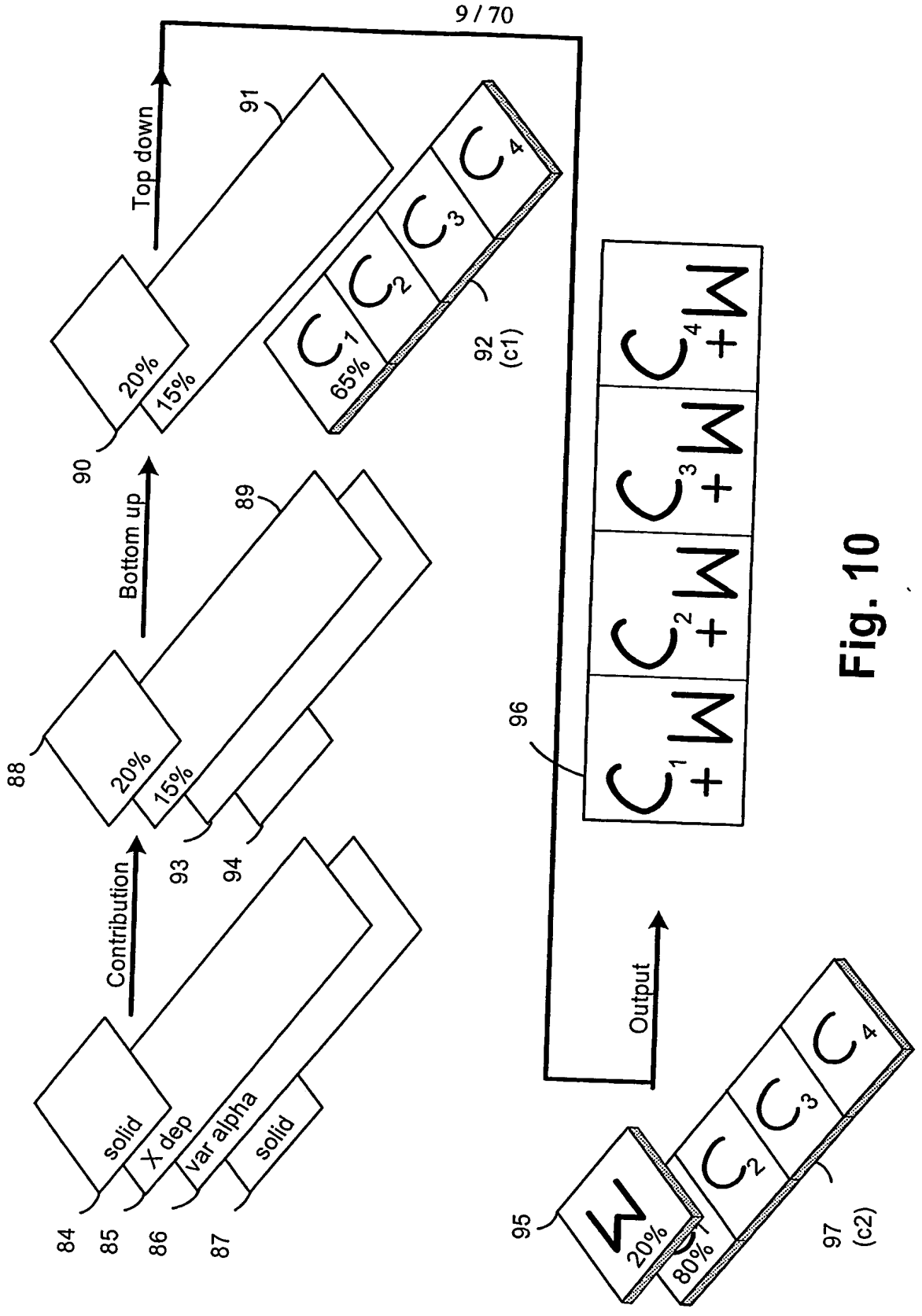

Top down
Bottom up
Contribution
Output
solid
X dep
var alpha
solid
20%
15%
65%
20%
80%
84
85
86
87
88
89
90
91
92
(c1)
93
94
95
96
97
(c2)

Fig. 10

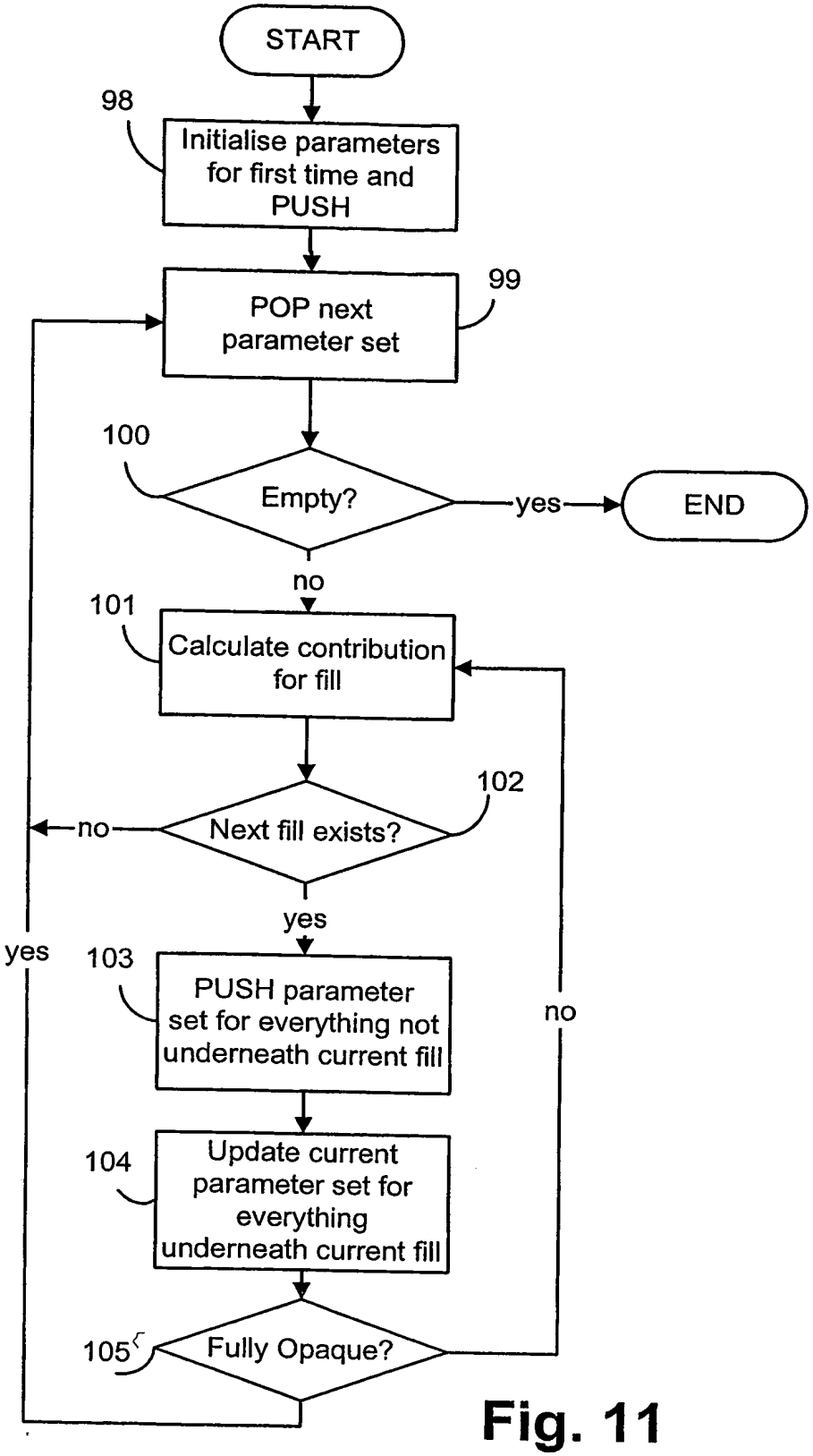
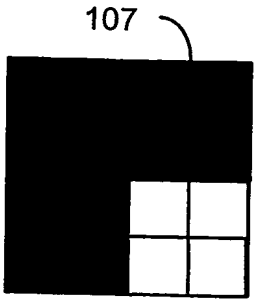
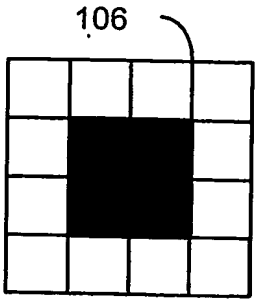

START
98
Initialise parameters for first time and PUSH
99
POP next parameter set
100
Empty?
yes
END
no
101
Calculate contribution for fill
102
Next fill exists?
no
yes
yes
103
PUSH parameter set for everything not underneath current fill
no
104
Update current parameter set for everything underneath current fill
105
Fully Opaque?

Fig. 11



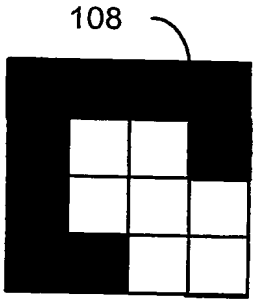
coverage

Fig. 12(a)



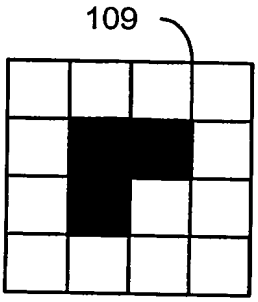
A-buffer

Fig. 12(b)



coverage ∩ ~(coverage ∩ A-buffer)

Fig. 12(c)



coverage ∩ A-buffer

Fig. 12(d)

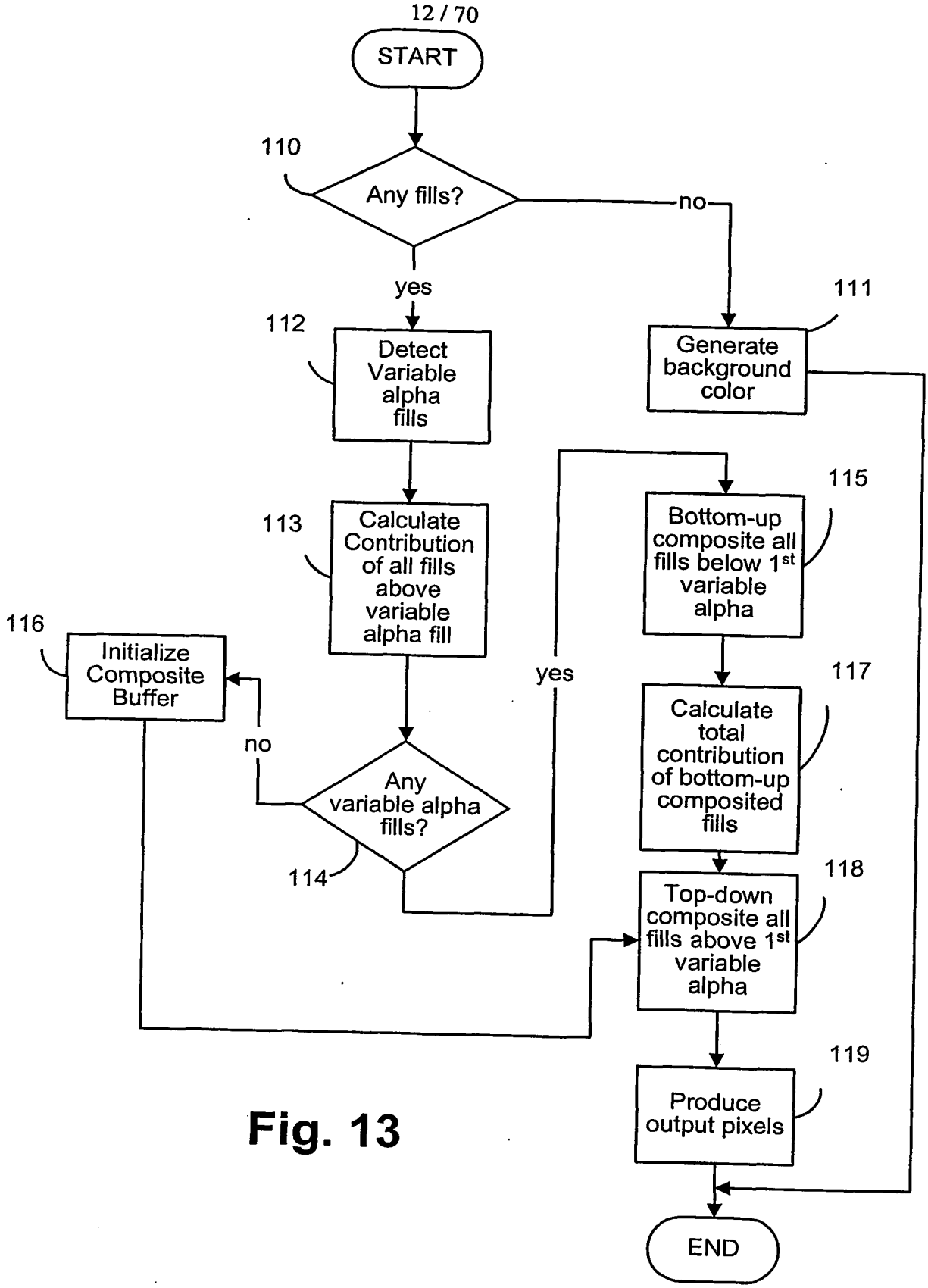

START
110
Any fills?
no
yes
111
Generate background color
112
Detect Variable alpha fills
113
Calculate Contribution of all fills above variable alpha fill
115
Bottom-up composite all fills below 1st variable alpha
116
Initialize Composite Buffer
no
Any variable alpha fills?
yes
114
117
Calculate total contribution of bottom-up composited fills
118
Top-down composite all fills above 1st variable alpha
119
Produce output pixels
END

Fig. 13

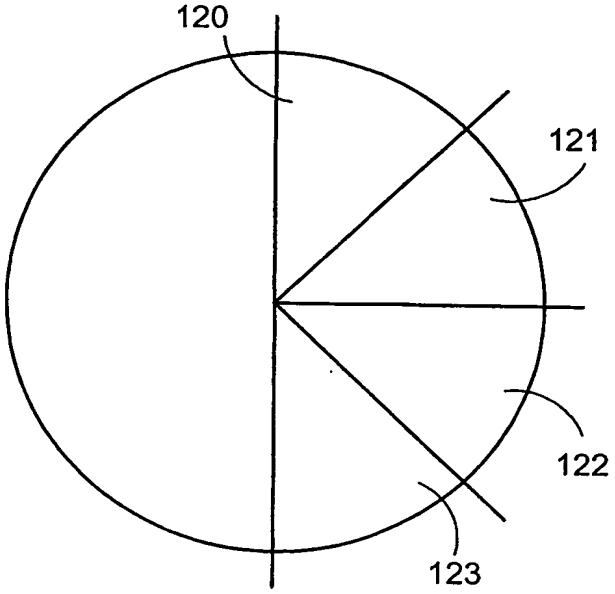


Fig. 14

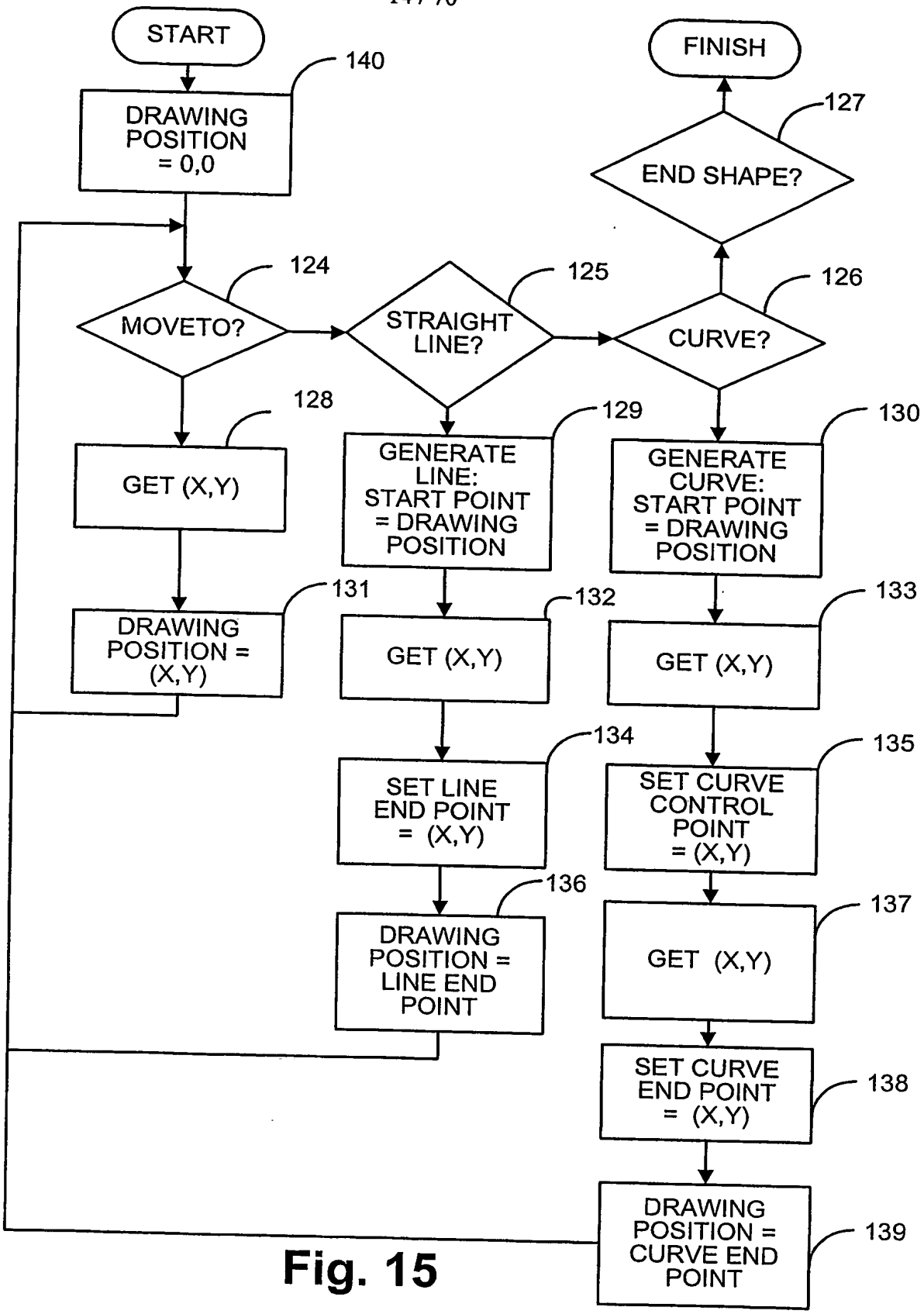

START
140
DRAWING POSITION = 0,0
FINISH
127
END SHAPE?
124
MOVETO?
125
STRAIGHT LINE?
126
CURVE?
128
GET (X,Y)
129
GENERATE LINE: START POINT = DRAWING POSITION
130
GENERATE CURVE: START POINT = DRAWING POSITION
131
DRAWING POSITION = (X,Y)
132
GET (X,Y)
133
GET (X,Y)
134
SET LINE END POINT = (X,Y)
135
SET CURVE CONTROL POINT = (X,Y)
136
DRAWING POSITION = LINE END POINT
137
GET (X,Y)
SET CURVE END POINT = (X,Y)
138
DRAWING POSITION = CURVE END POINT
139

Fig. 15

Fig. 16(a)

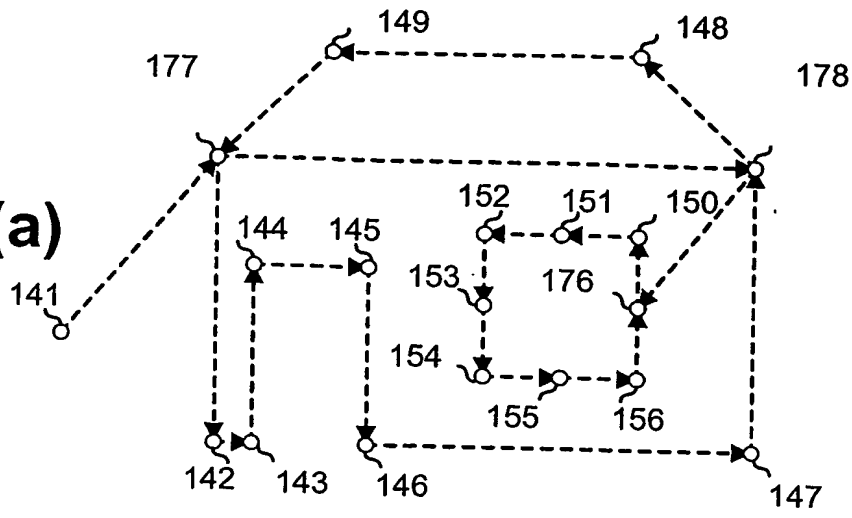


Fig. 16(b)

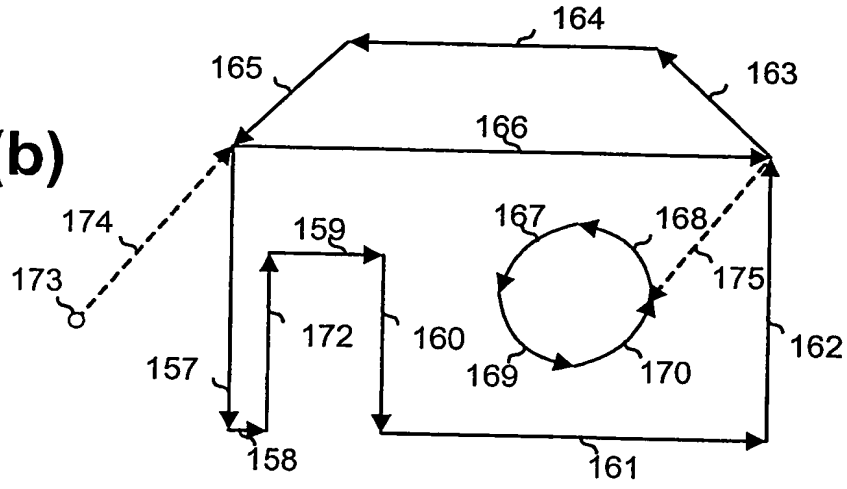
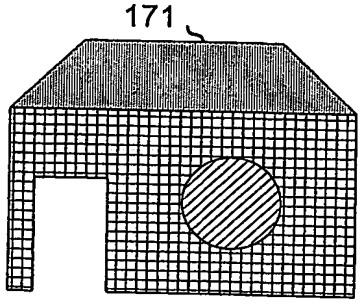


Fig. 16(c)



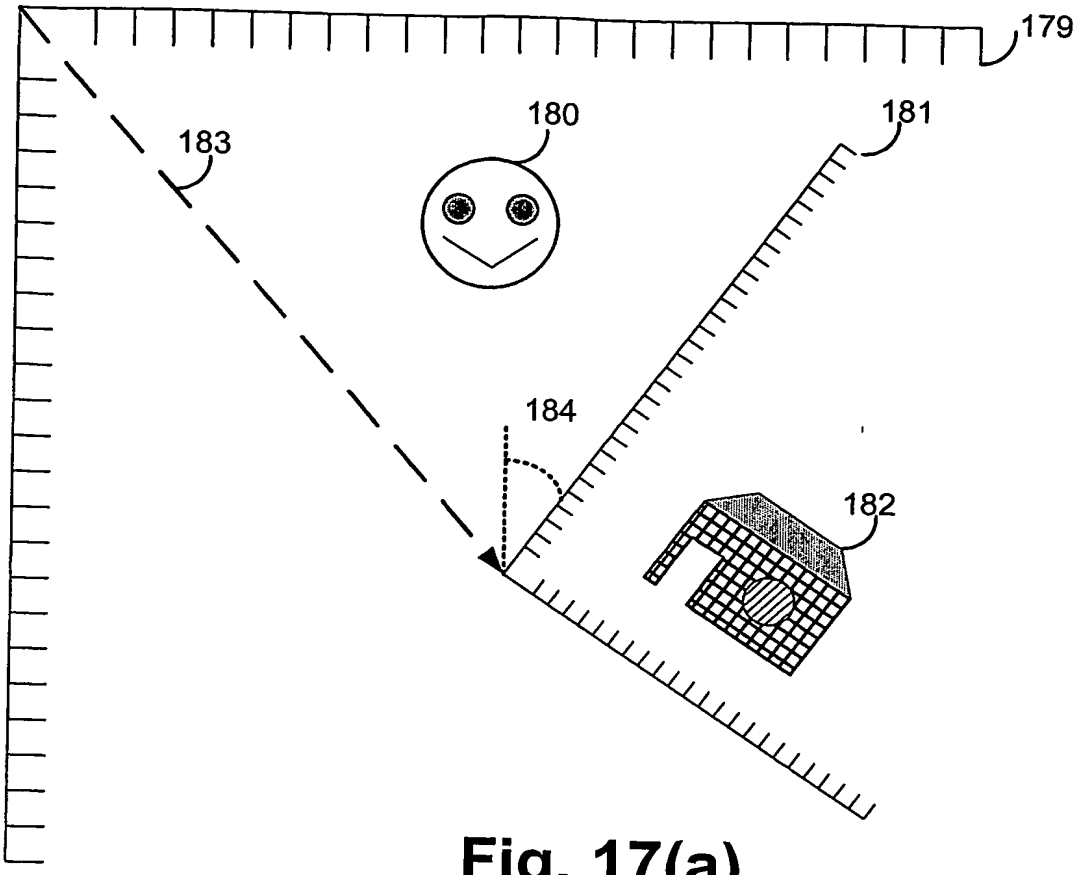


Fig. 17(a)

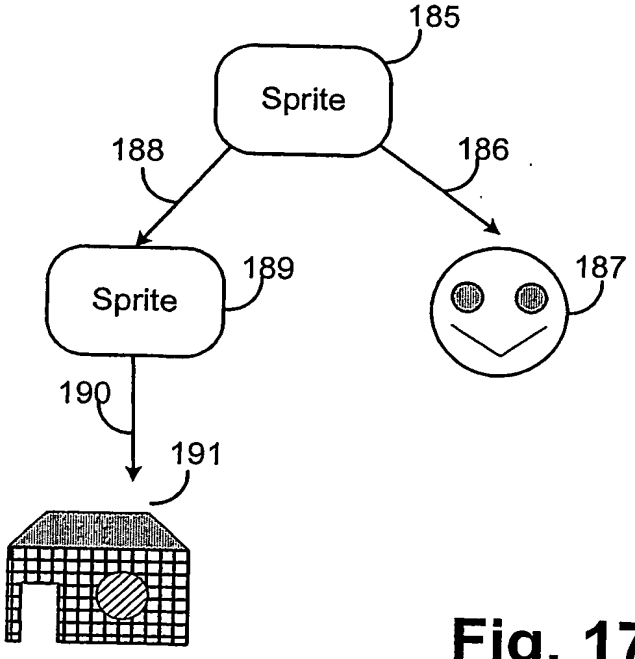


Fig. 17(b)

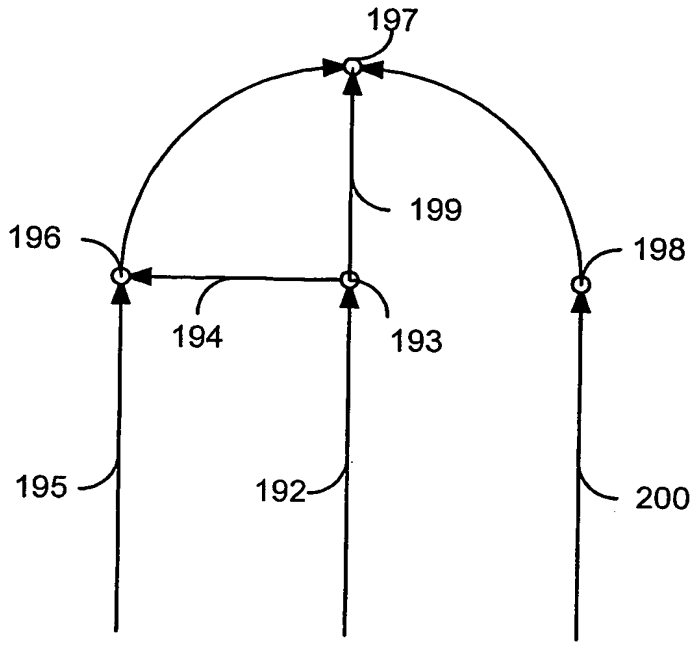


Fig. 18(a)

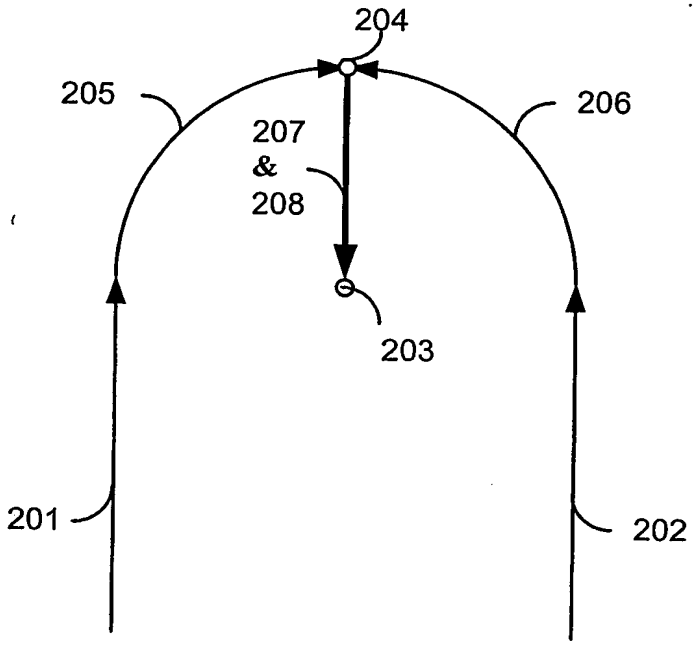


Fig. 18(b)

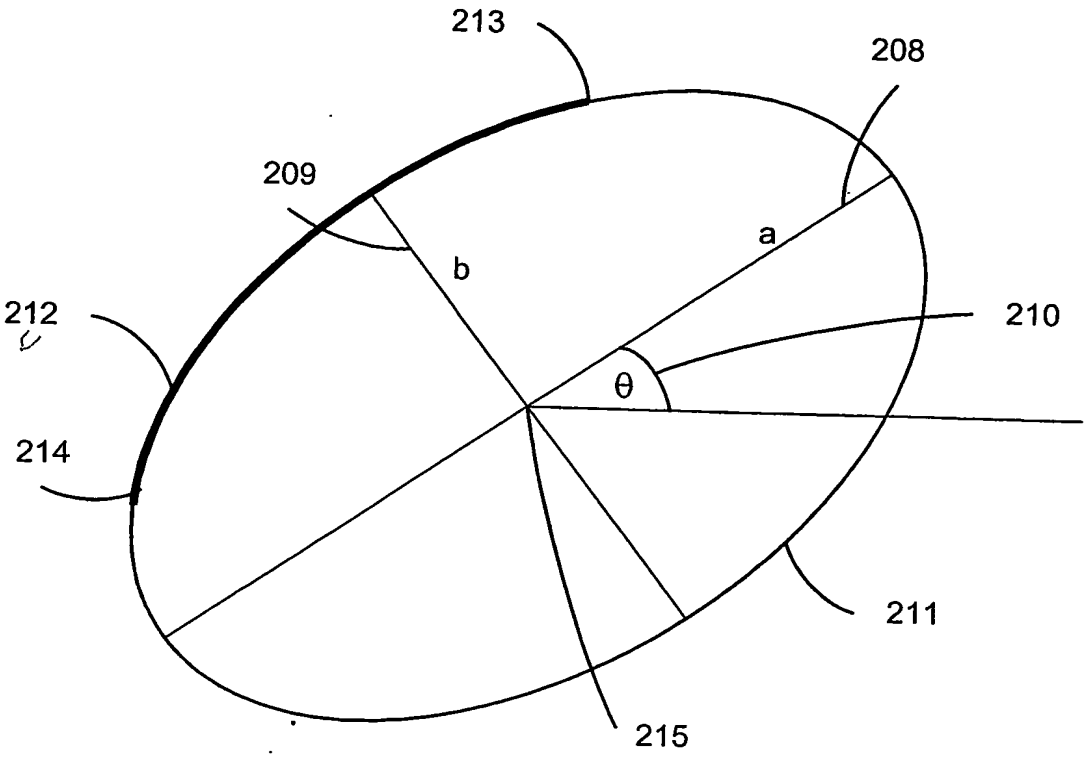


Fig. 19

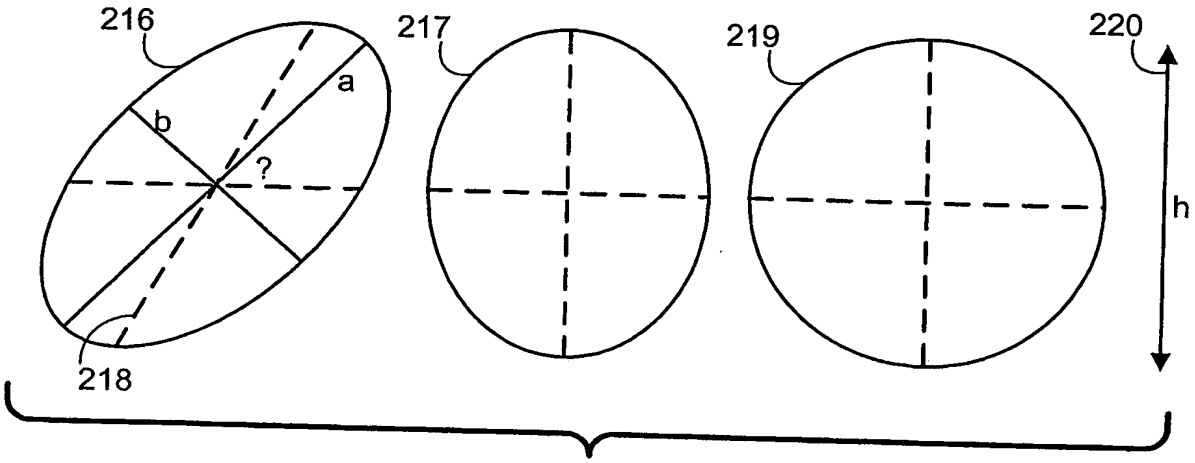


Fig. 20

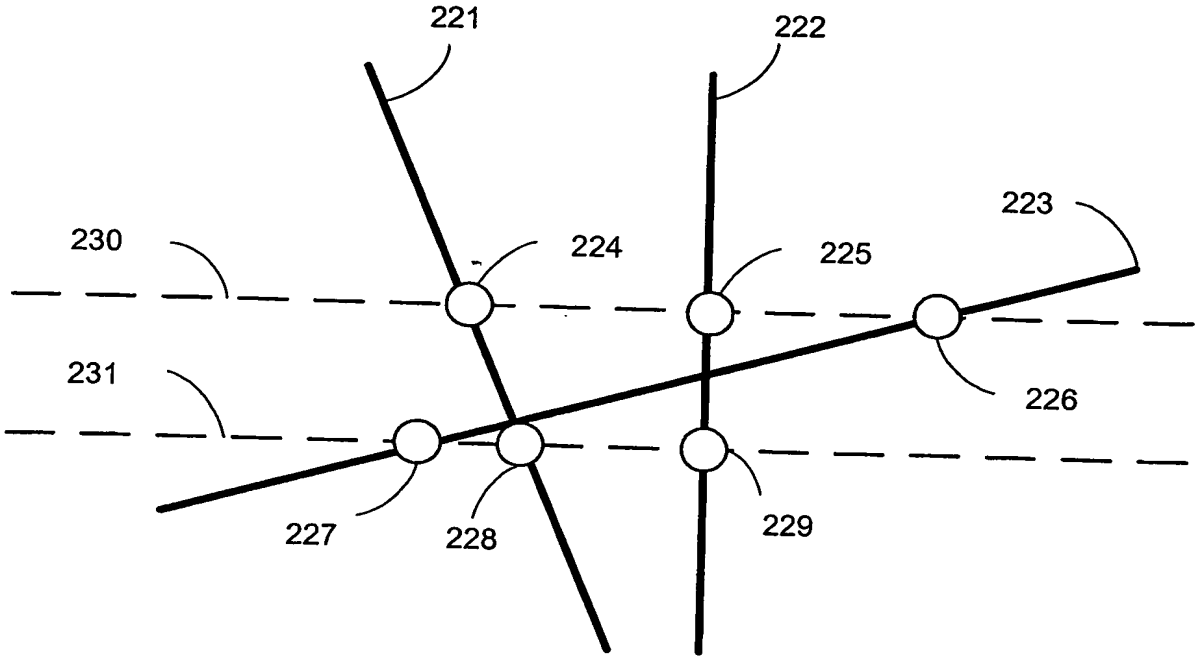


Fig. 21

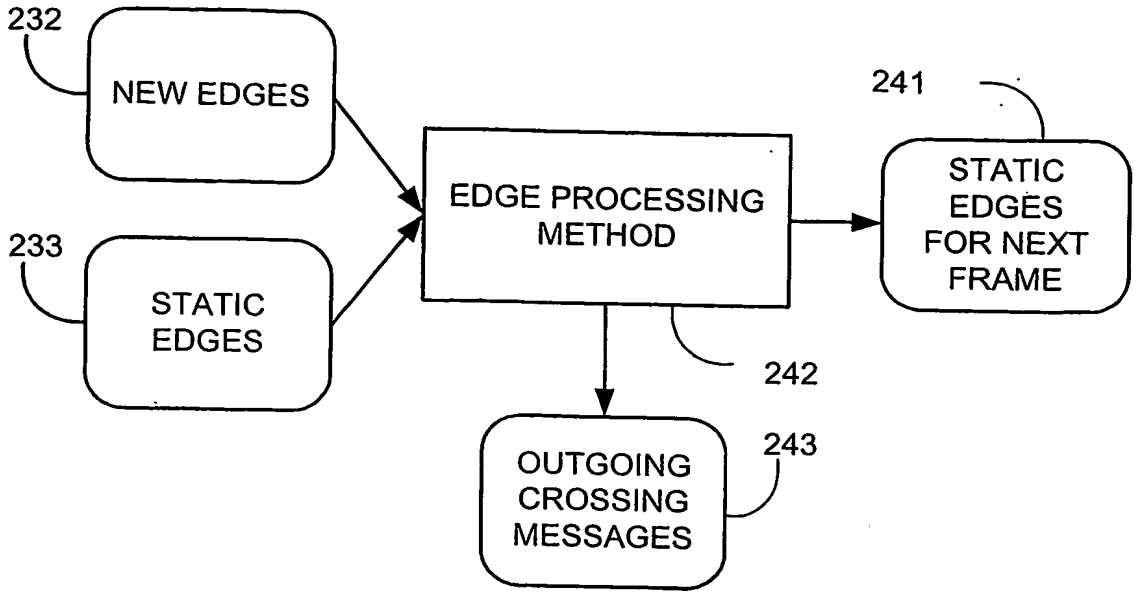


Fig. 22(a)

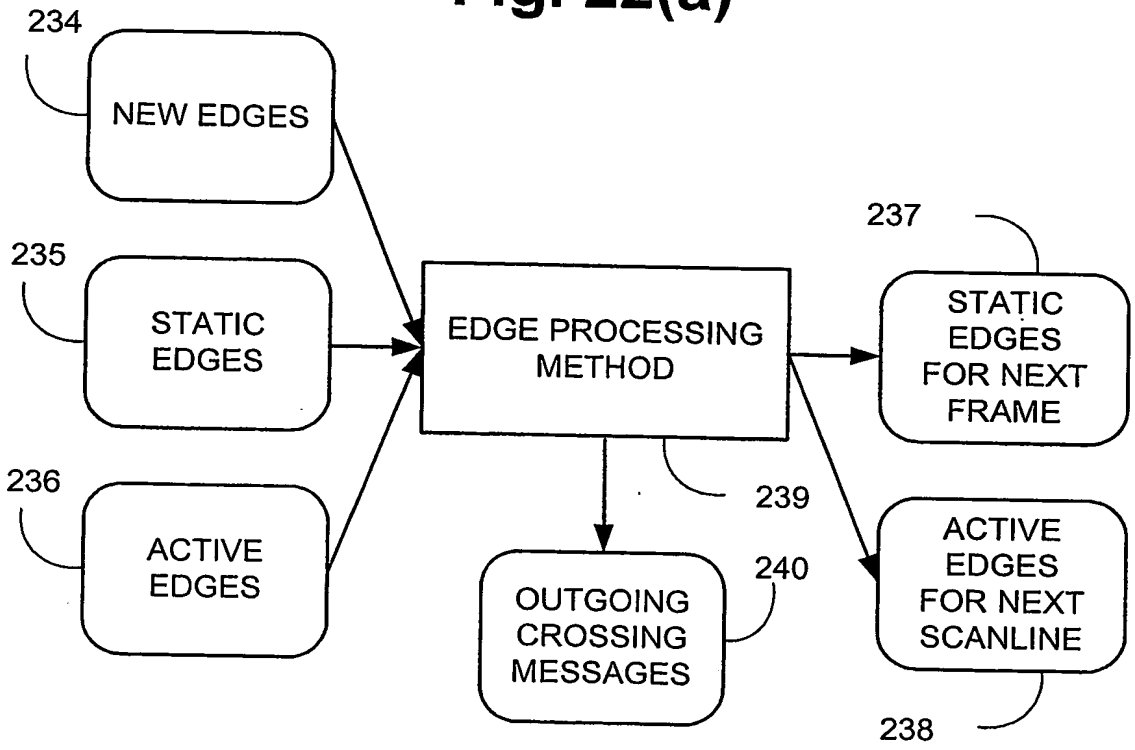


Fig. 22(b)

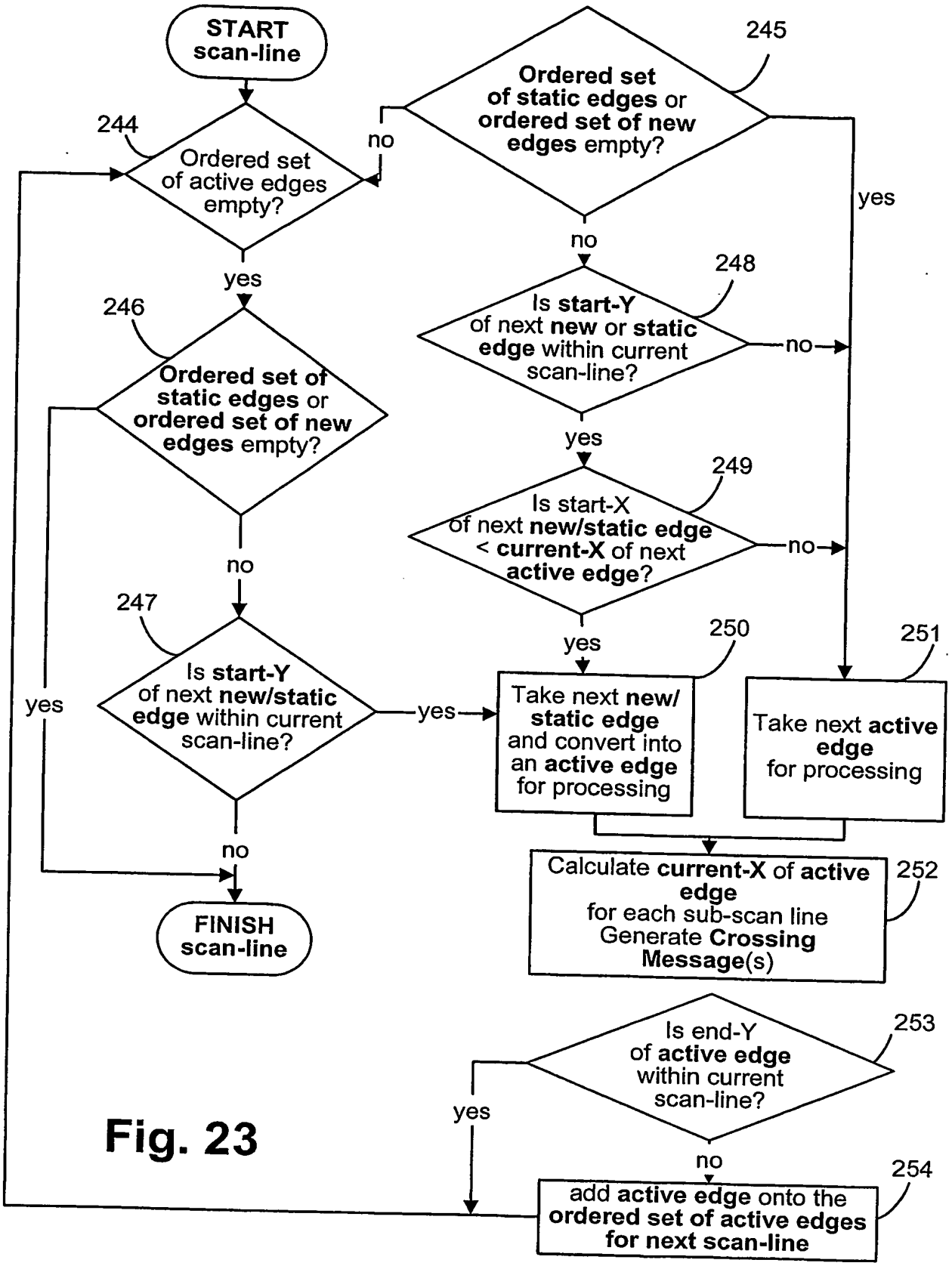

START scan-line
244
Ordered set of active edges empty?
no
yes
246
Ordered set of static edges or ordered set of new edges empty?
no
247
Is start-Y of next new/static edge within current scan-line?
yes
yes
no
FINISH scan-line
245
Ordered set of static edges or ordered set of new edges empty?
yes
no
248
Is start-Y of next new or static edge within current scan-line?
no
yes
249
Is start-X of next new/static edge < current-X of next active edge?
no
yes
250
Take next new/ static edge and convert into an active edge for processing
251
Take next active edge for processing
Calculate current-X of active edge for each sub-scan line Generate Crossing Message(s)
252
253
Is end-Y of active edge within current scan-line?
yes
no
254
add active edge onto the ordered set of active edges for next scan-line

Fig. 23

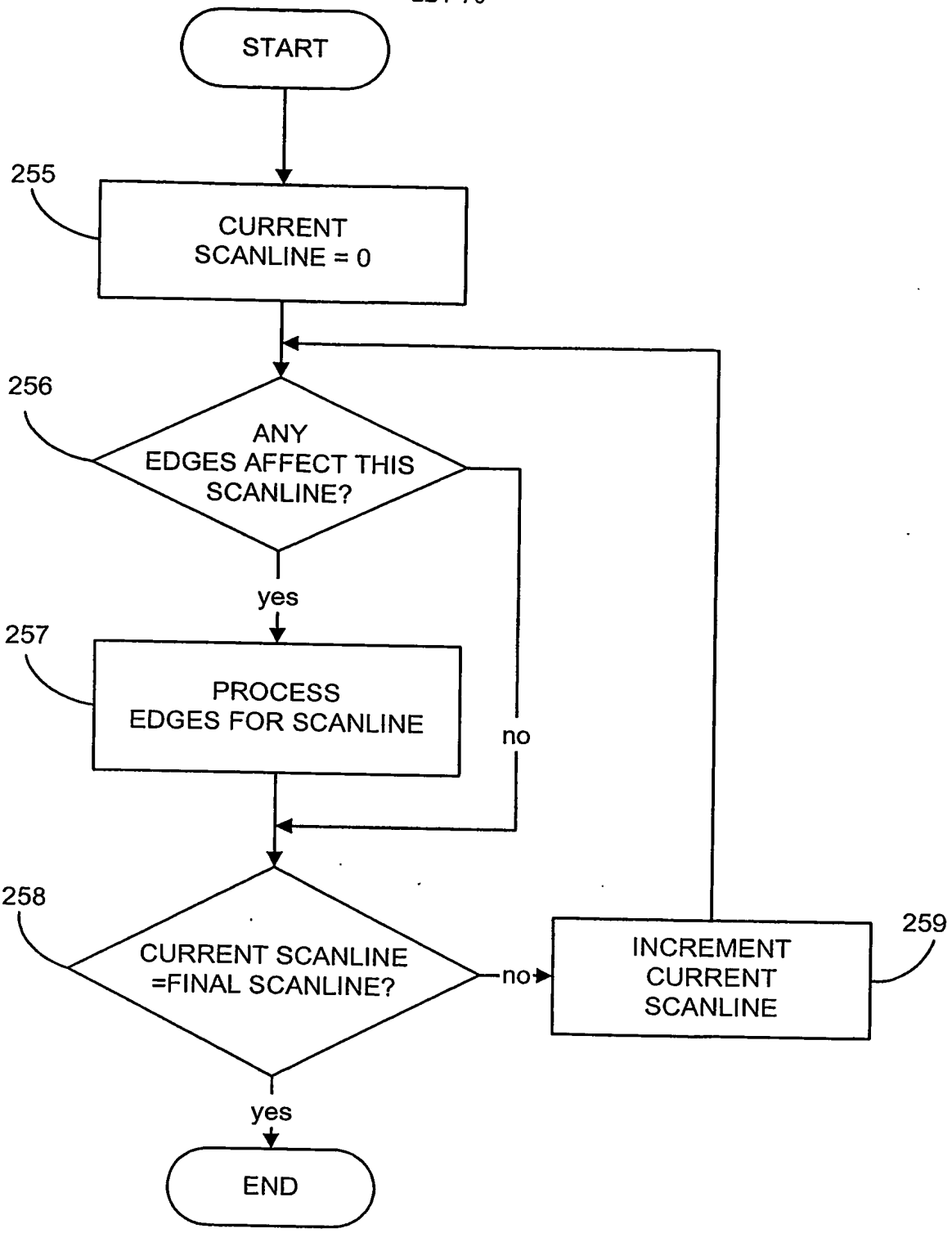

START
255
CURRENT SCANLINE = 0
256
ANY EDGES AFFECT THIS SCANLINE?
yes
no
257
PROCESS EDGES FOR SCANLINE
258
CURRENT SCANLINE =FINAL SCANLINE?
no
259
INCREMENT CURRENT SCANLINE
yes
END

Fig. 24

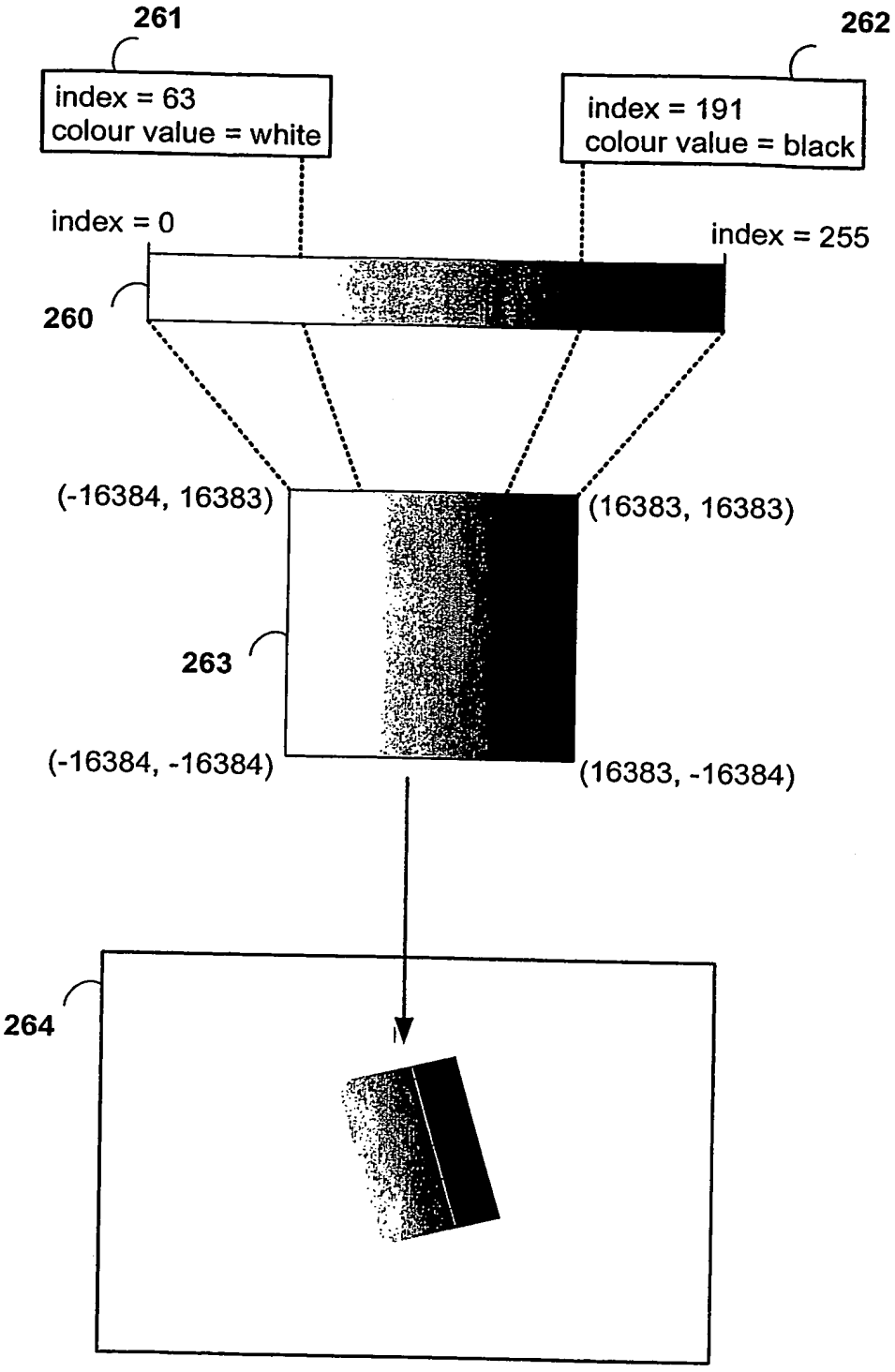

261
index = 63
colour value = white
262
index = 191
colour value = black
index = 0
index = 255
260
(-16384, 16383)
(16383, 16383)
263
(-16384, -16384)
(16383, -16384)
264

Fig. 25

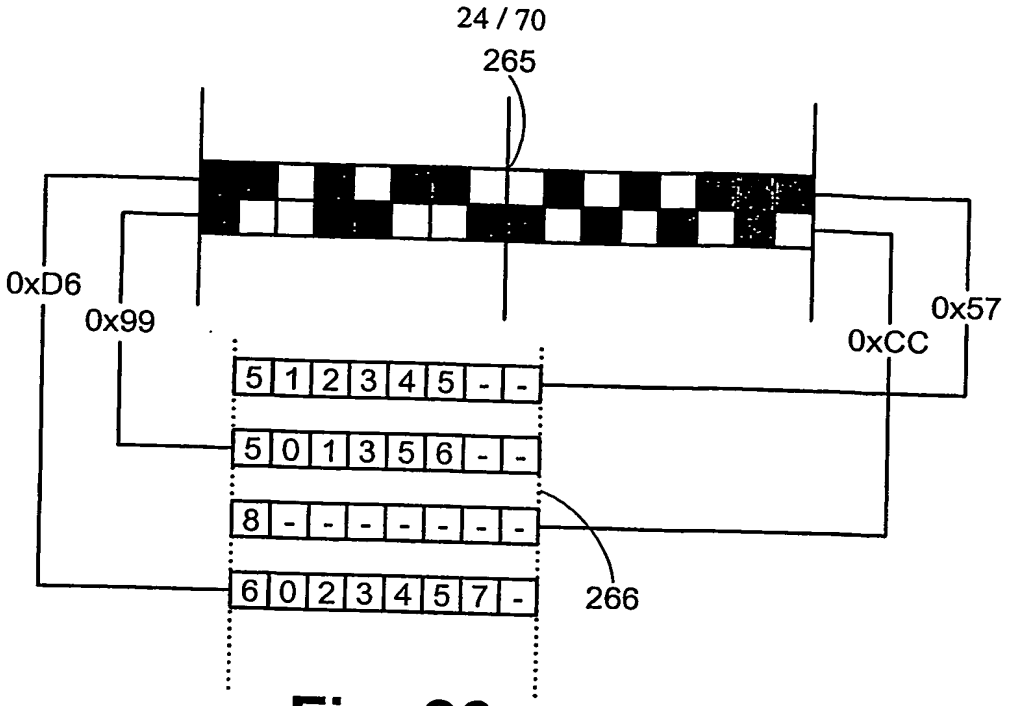

265
0xD6
0x99
0x57
0xCC
5 1 2 3 4 5 - -
5 0 1 3 5 6 - -
8 - - - - - - -
6 0 2 3 4 5 7 -
266

Fig. 26

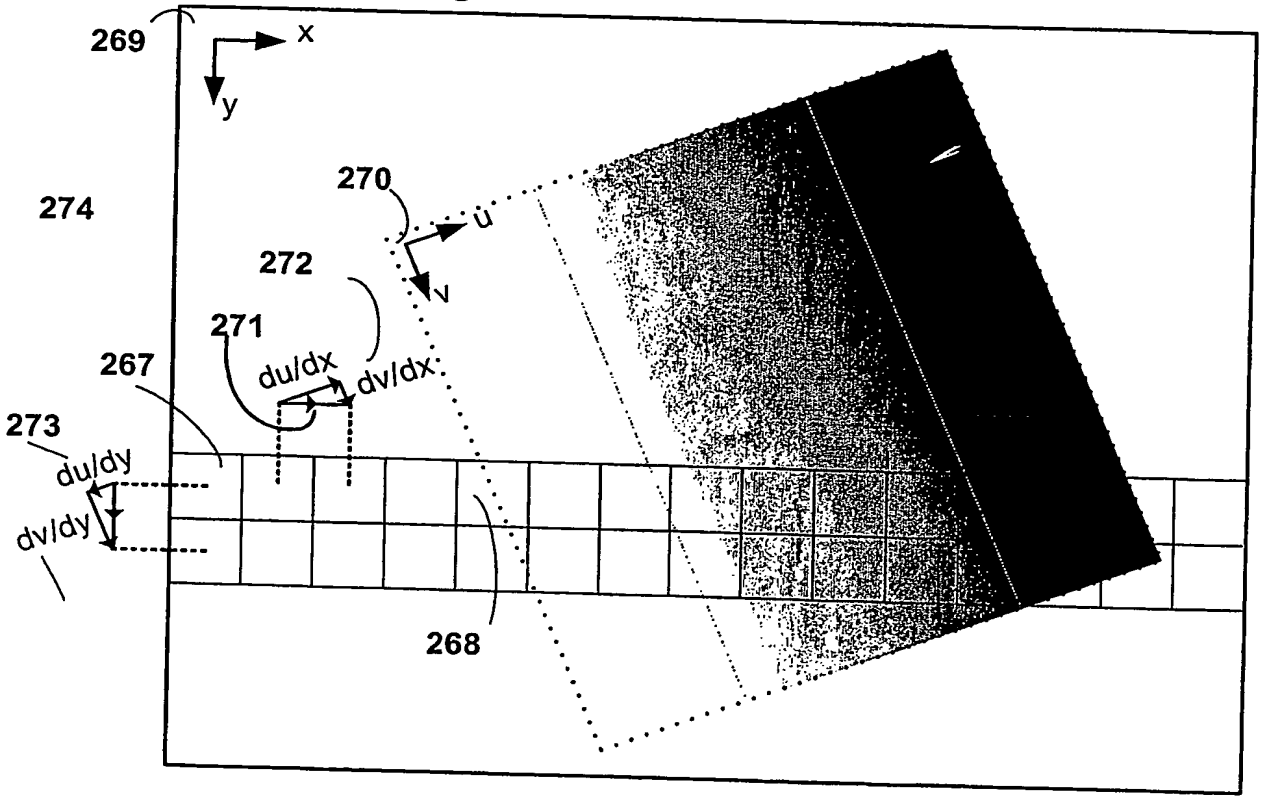

269
x
y
270
u
v
274
272
271
267
du/dx
dv/dx
273
du/dy
dv/dy
268

Fig. 27

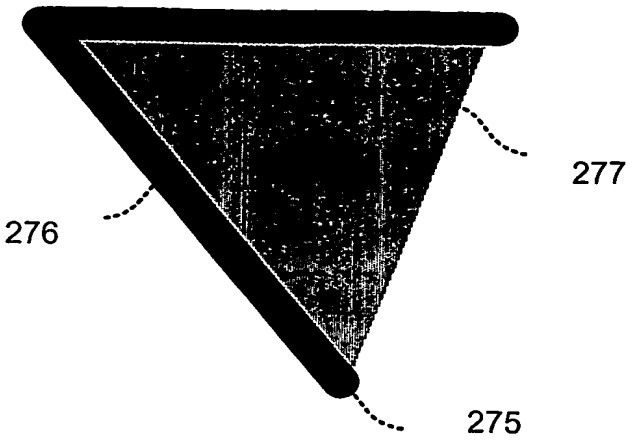


Fig. 28(a)

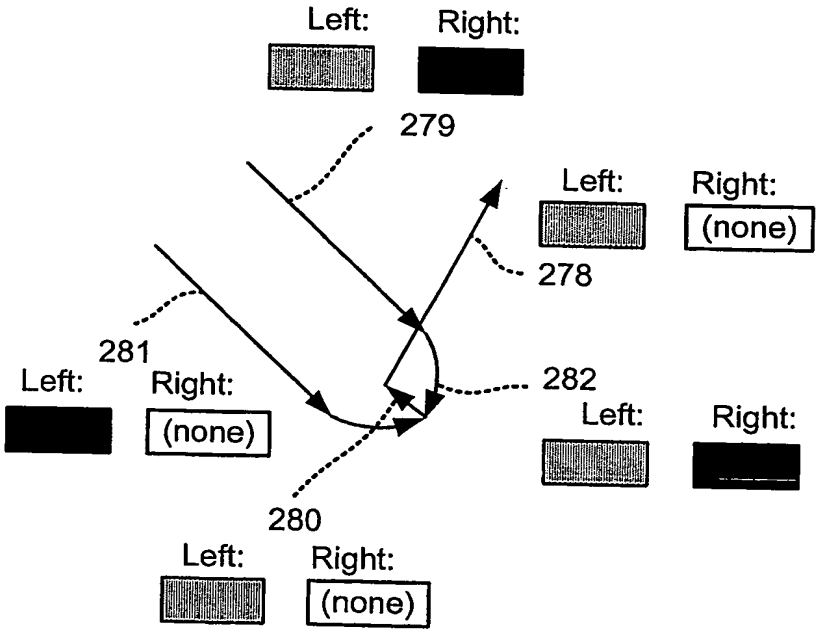


Fig. 28(b)

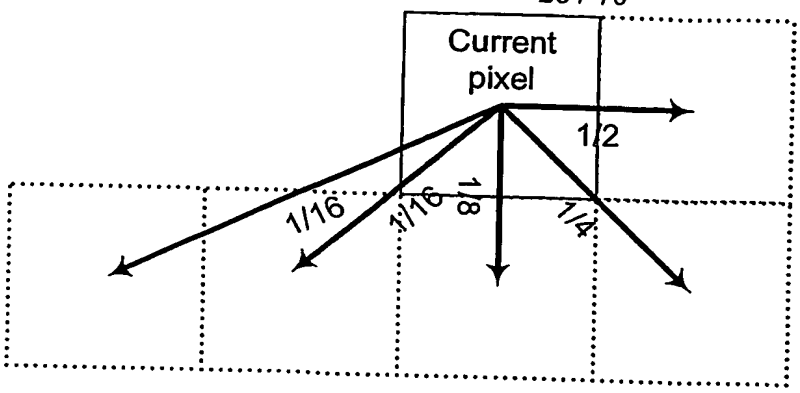


Fig. 29(a)

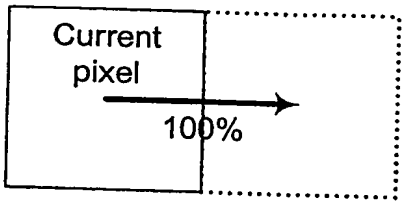


Fig. 29(b)

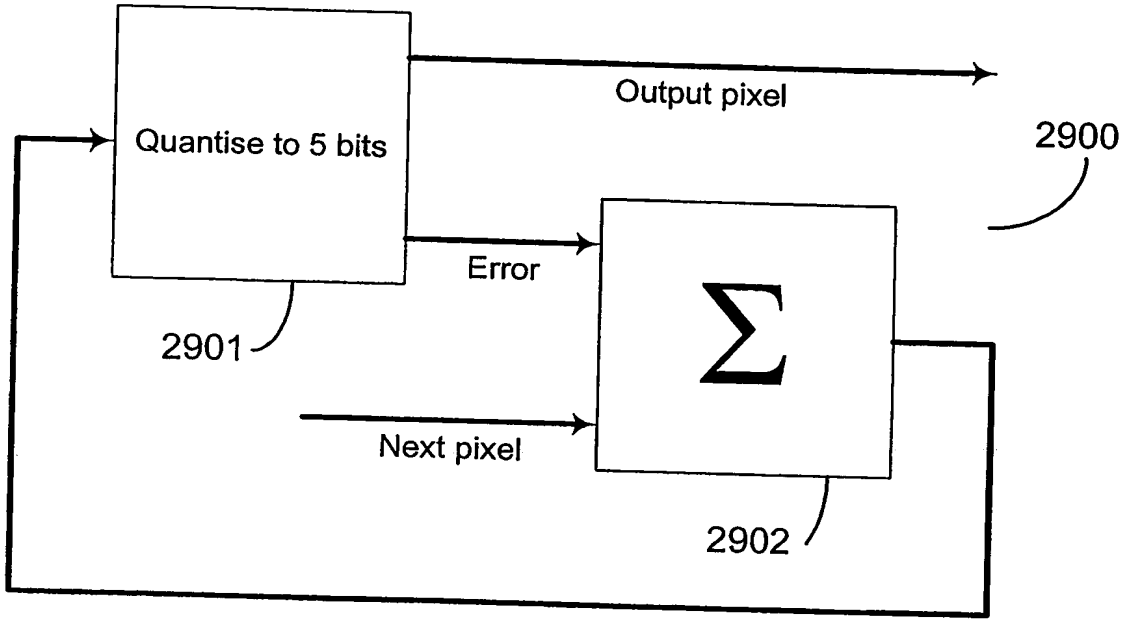


Fig. 29(c)

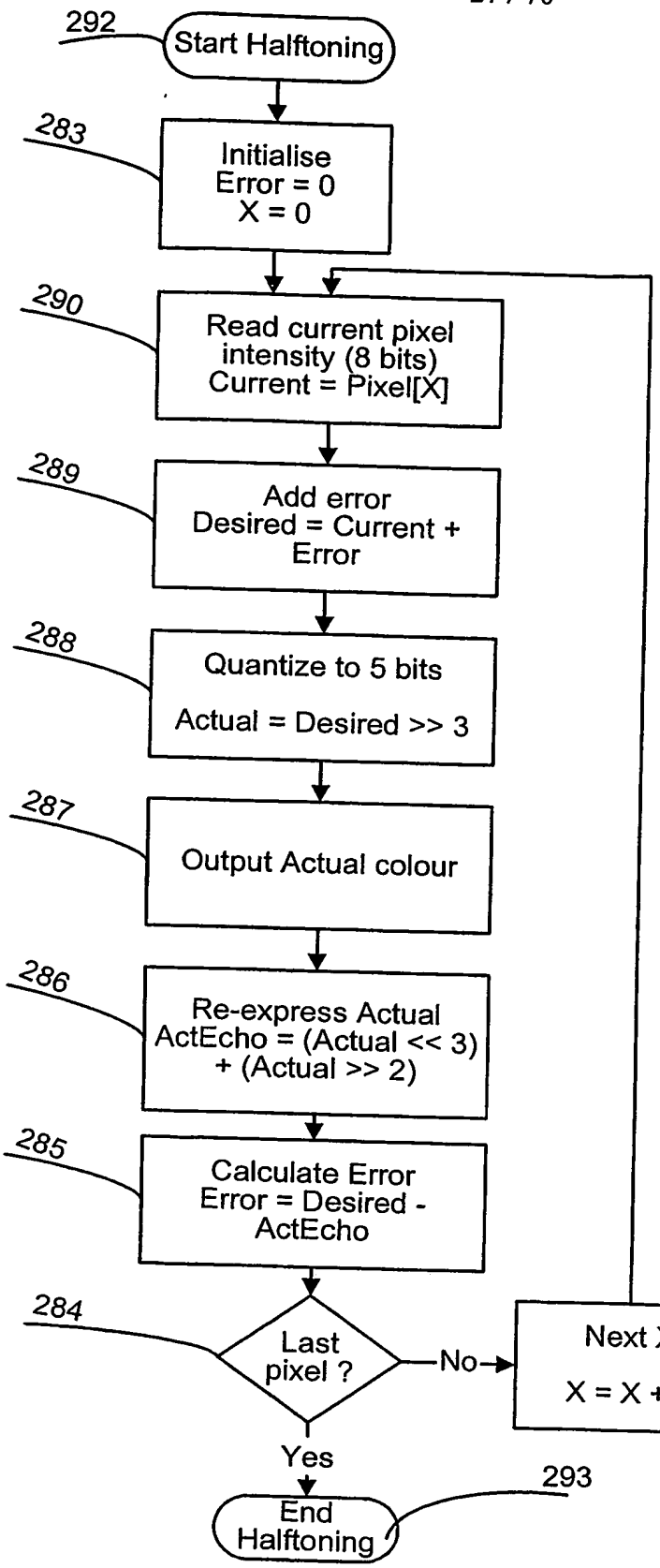

292
Start Halftoning
283
Initialise
Error = 0
X = 0
290
Read current pixel intensity (8 bits)
Current = Pixel[X]
289
Add error
Desired = Current + Error
288
Quantize to 5 bits
Actual = Desired >> 3
287
Output Actual colour
286
Re-express Actual
ActEcho = (Actual << 3) + (Actual >> 2)
285
Calculate Error
Error = Desired - ActEcho
284
Last pixel ?
No
Yes
291
Next X
X = X + 1
293
End Halftoning

Fig. 30

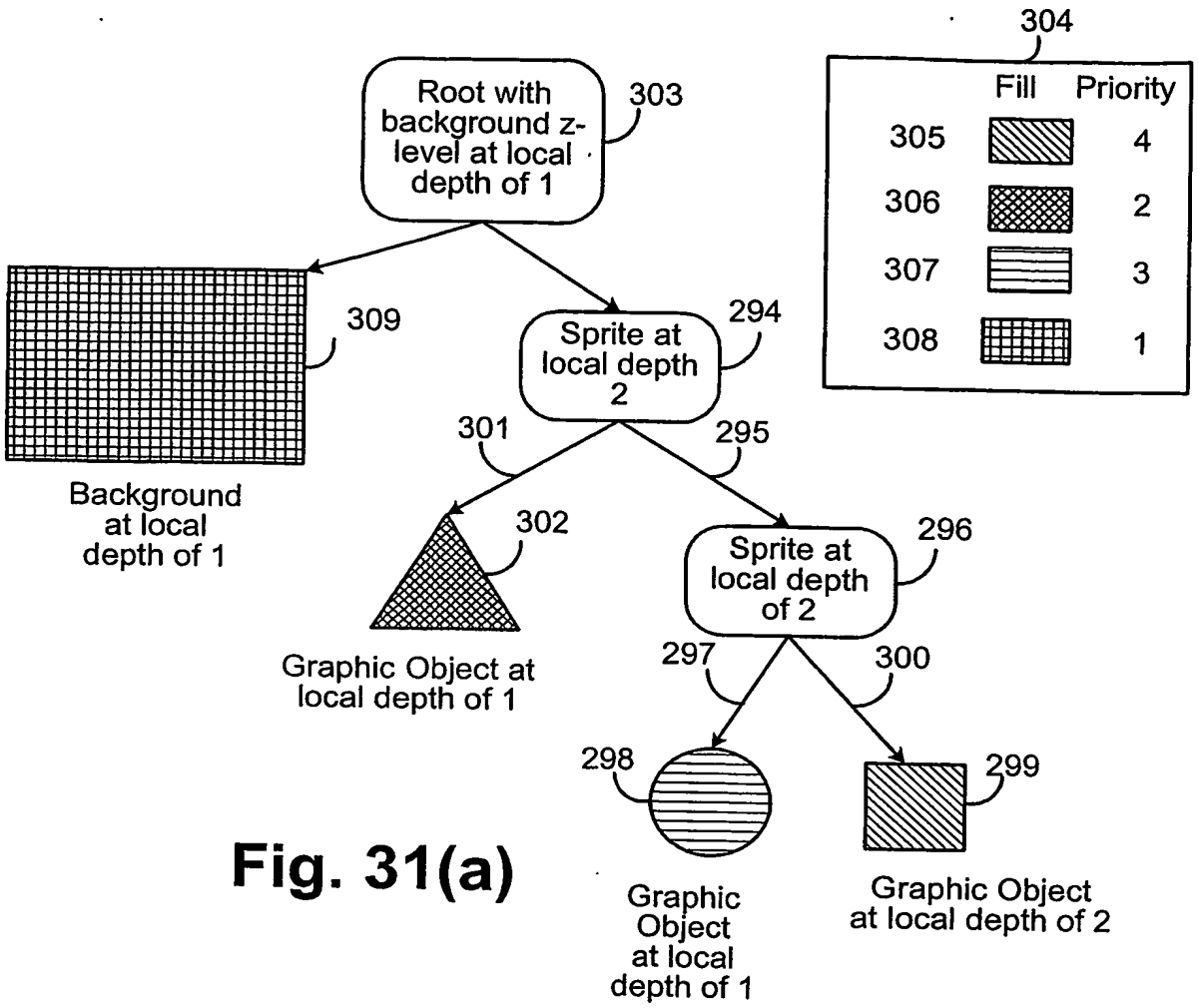


Fig. 31(a)

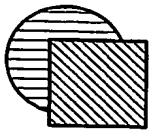


Fig. 31(b)

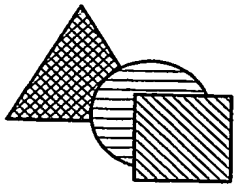


Fig. 31(c)

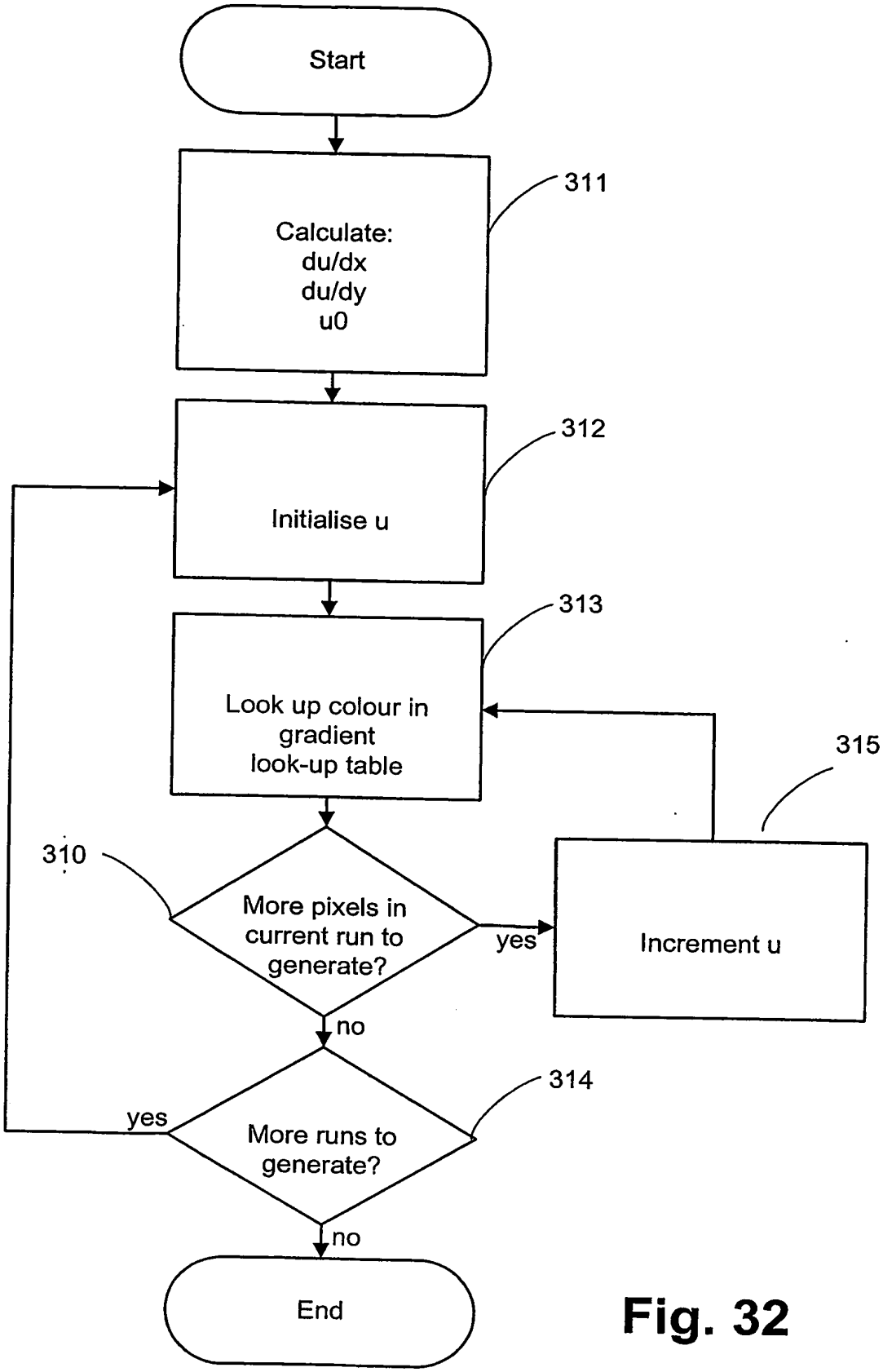

Start
Calculate:
du/dx
du/dy
u0
311
Initialise u
312
Look up colour in
gradient
look-up table
313
315
310
More pixels in
current run to
generate?
yes
Increment u
no
314
yes
More runs to
generate?
no
End

Fig. 32

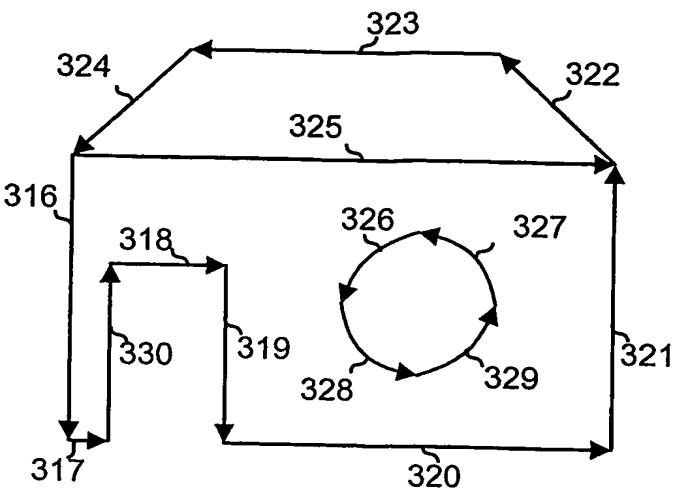
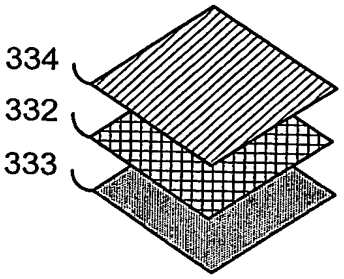
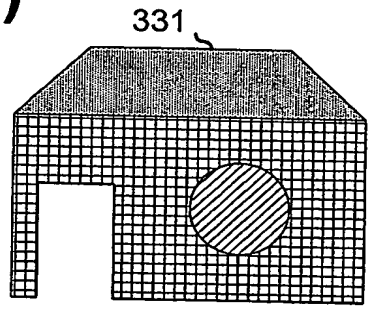


Fig. 33(a)



z-levels in order of priority

Fig. 33(b)



Rendered result

Fig. 33(c)

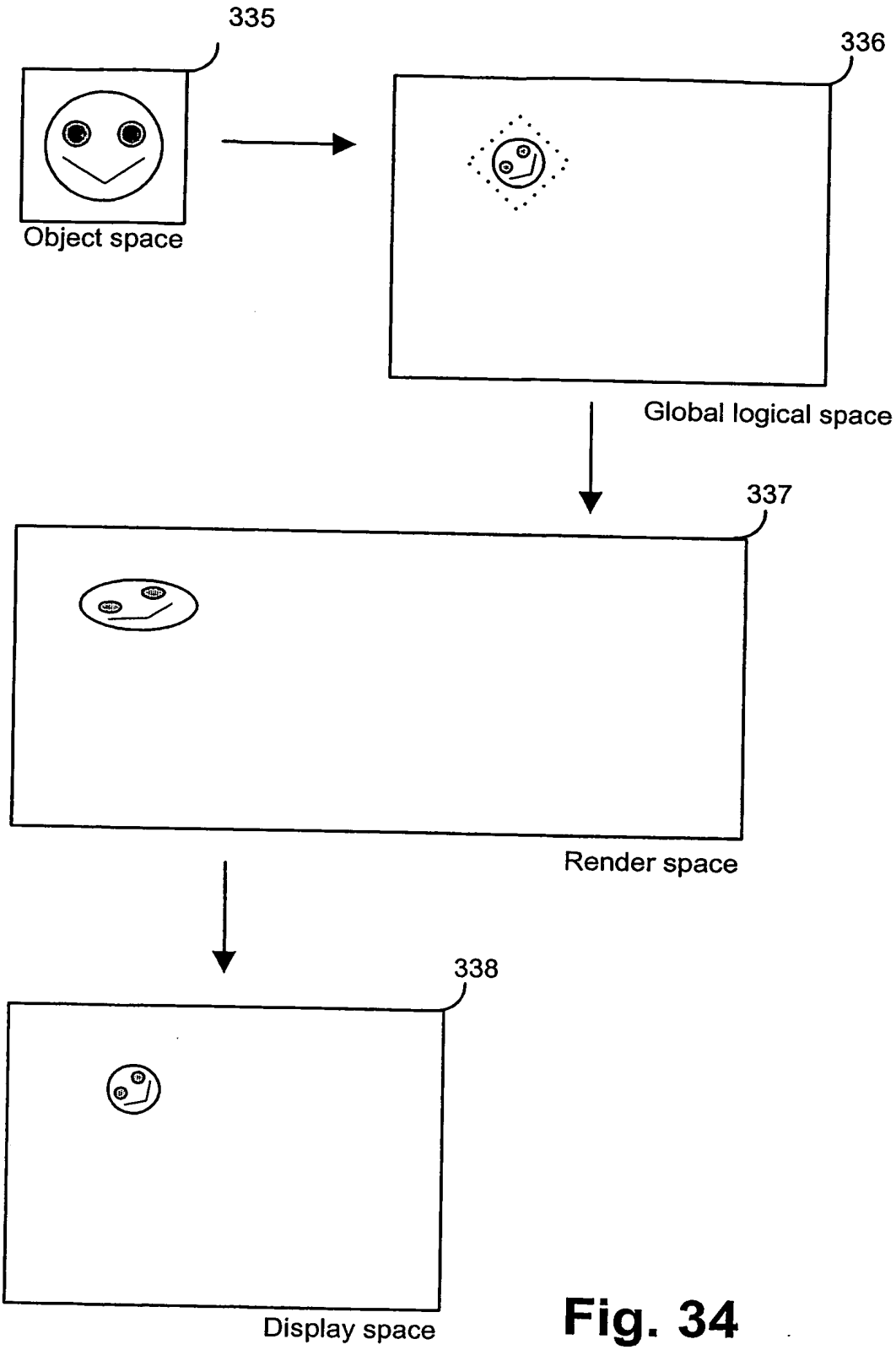

335
336
Object space
Global logical space
337
Render space
338
Display space

Fig. 34

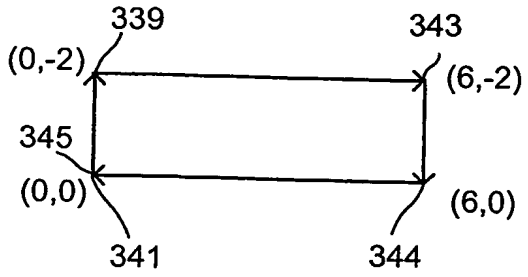


Fig. 35(a)

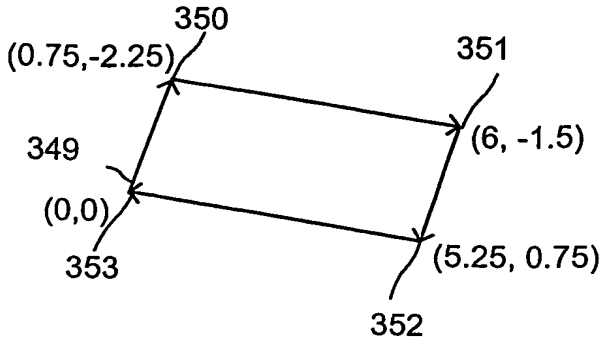


Fig. 35(b)

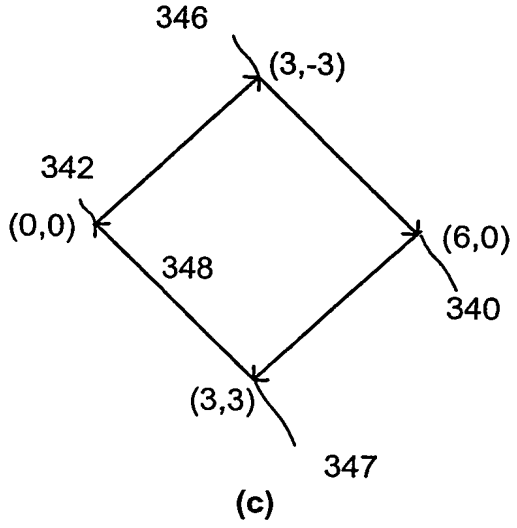


Fig. 35(c)

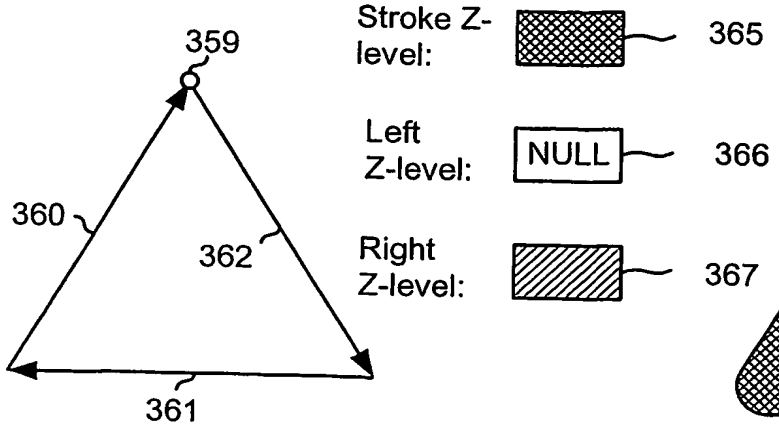


Fig. 36(a)

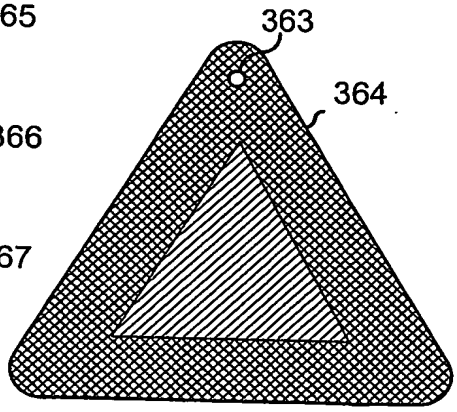


Fig. 36(b)

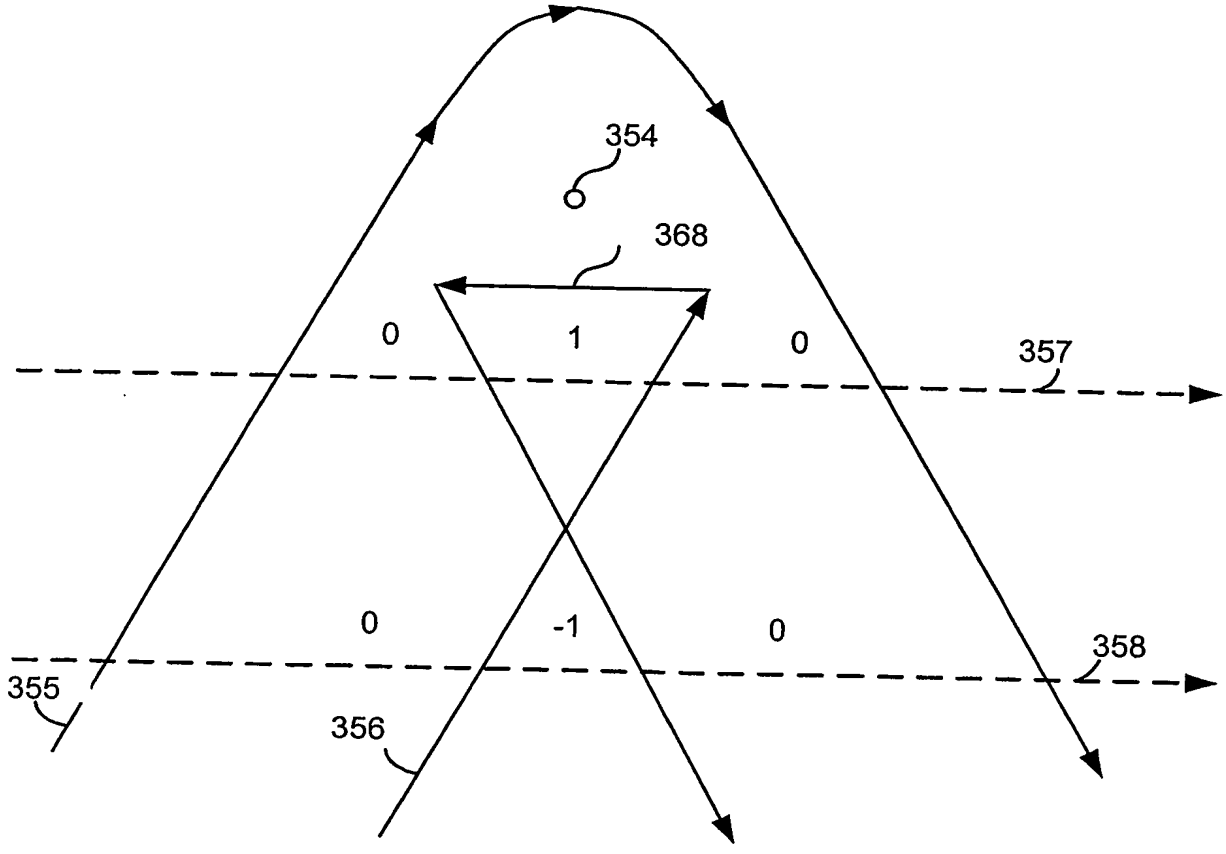
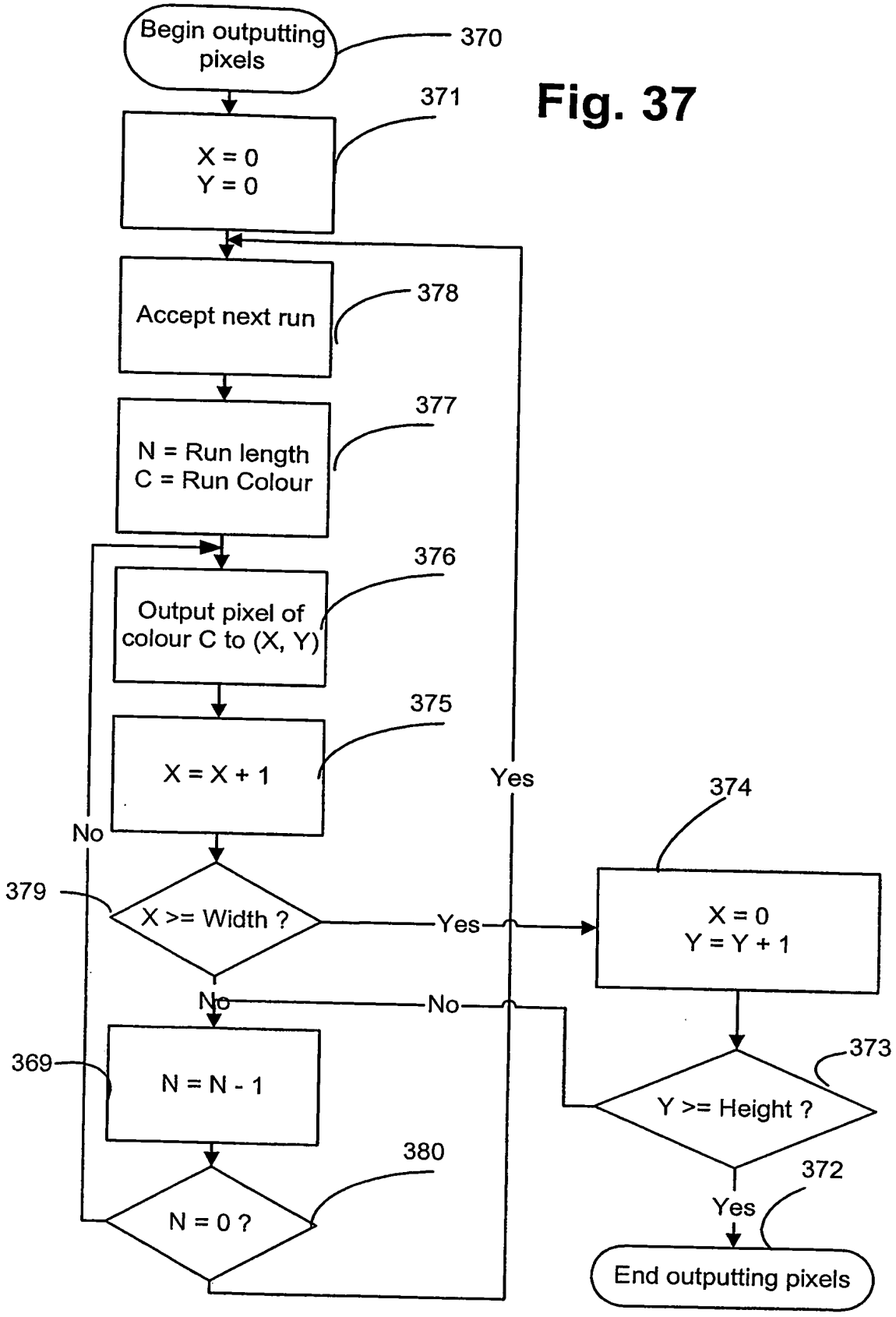


Fig. 36(c)

Fig. 37


Begin outputting pixels
370
371
X = 0
Y = 0
Accept next run
378
377
N = Run length
C = Run Colour
376
Output pixel of colour C to (X, Y)
375
X = X + 1
379
X >= Width ?
Yes
No
374
X = 0
Y = Y + 1
No
373
Y >= Height ?
369
N = N - 1
380
N = 0 ?
Yes
372
End outputting pixels

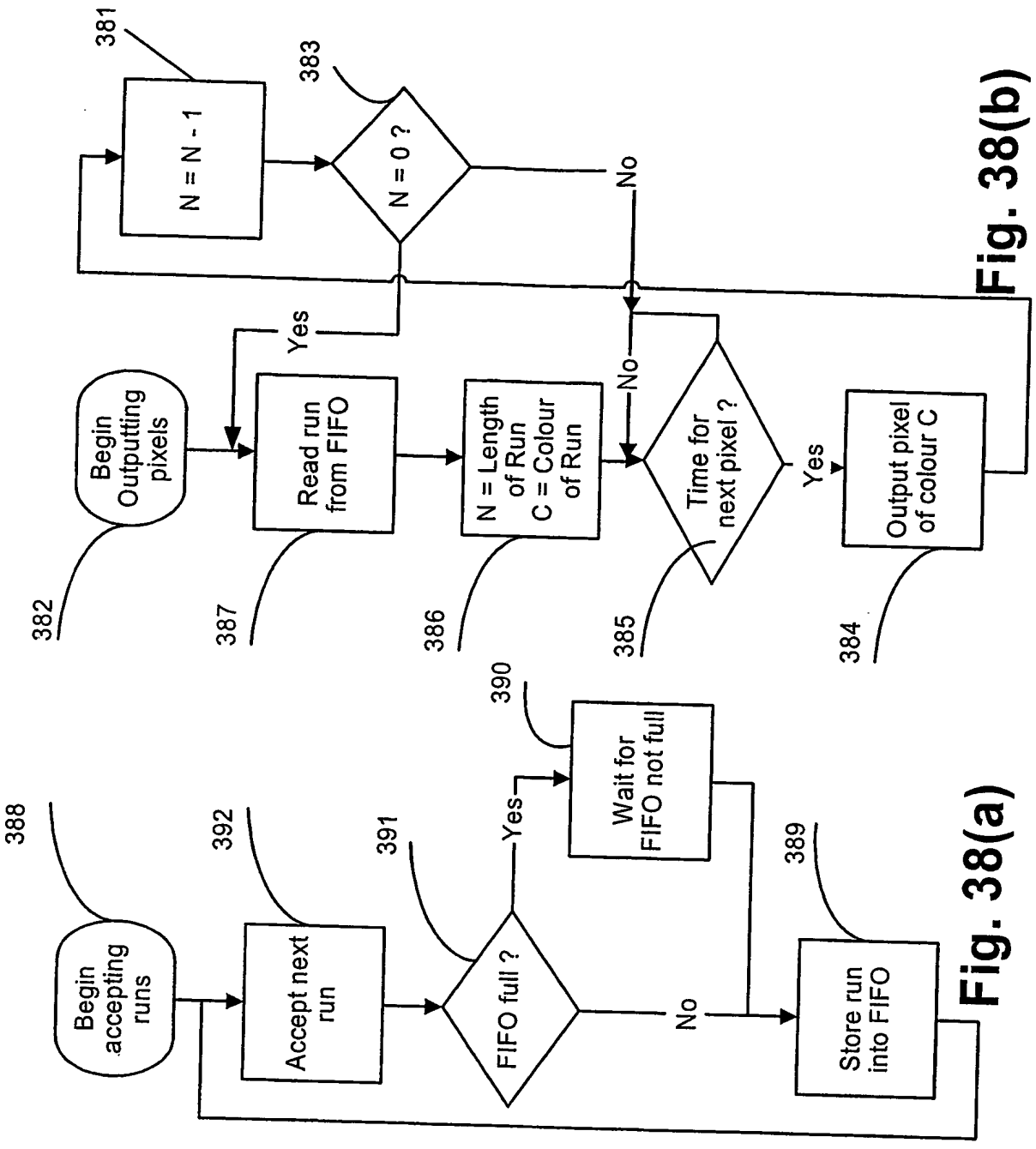


Fig. 38(a)

Fig. 38(b)

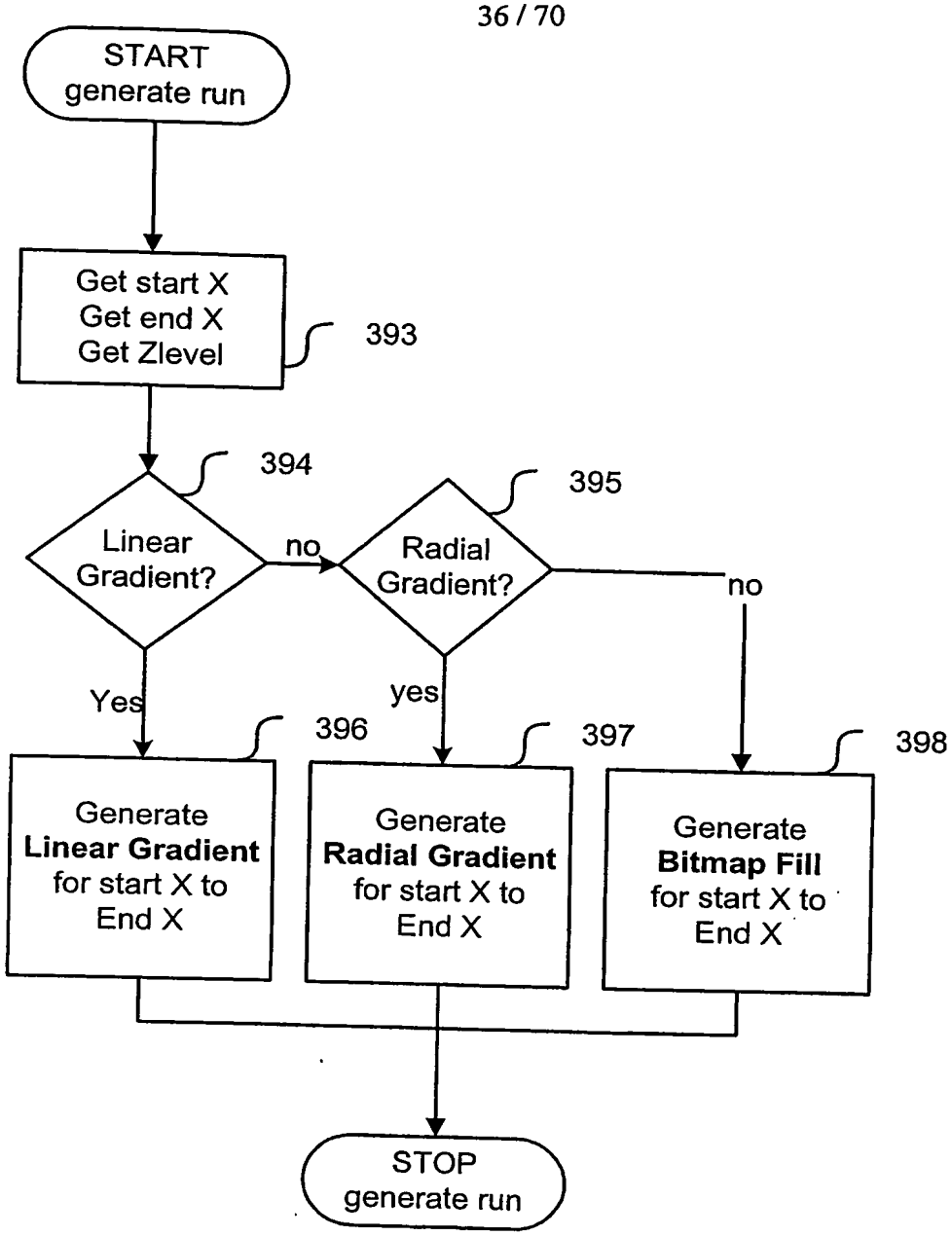

START
generate run
Get start X
Get end X
Get Zlevel
393
394
Linear
Gradient?
no
395
Radial
Gradient?
no
Yes
yes
396
397
398
Generate
Linear Gradient
for start X to
End X
Generate
Radial Gradient
for start X to
End X
Generate
Bitmap Fill
for start X to
End X
STOP
generate run

Fig. 39

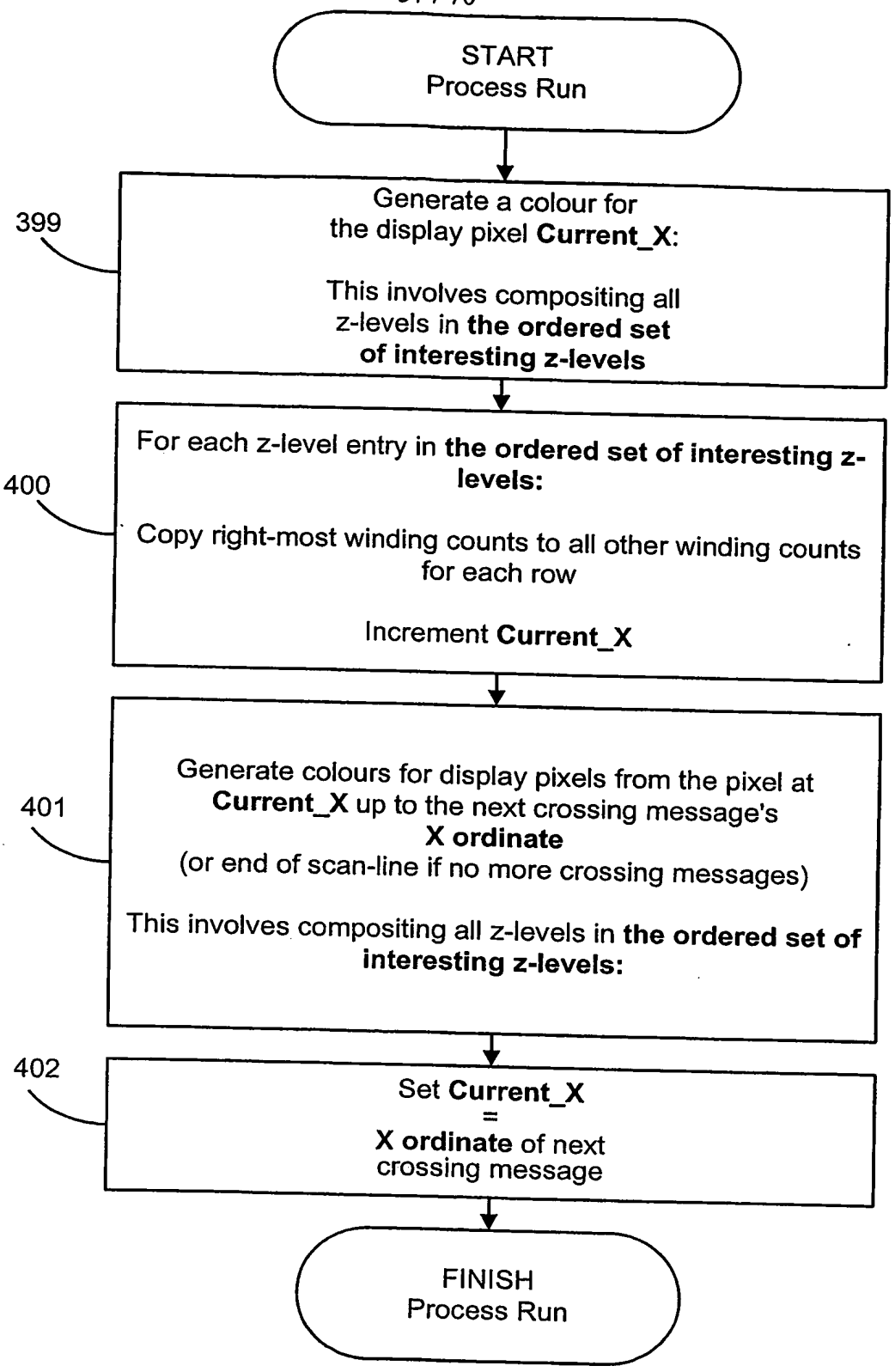

START
Process Run
399
Generate a colour for the display pixel **Current_X**:
This involves compositing all z-levels in **the ordered set of interesting z-levels**
400
For each z-level entry in **the ordered set of interesting z-levels:**
Copy right-most winding counts to all other winding counts for each row
Increment **Current_X**
401
Generate colours for display pixels from the pixel at **Current_X** up to the next crossing message's **X ordinate**
(or end of scan-line if no more crossing messages)
This involves compositing all z-levels in **the ordered set of interesting z-levels:**
402
Set **Current_X** = **X ordinate** of next crossing message
FINISH
Process Run

Fig. 40

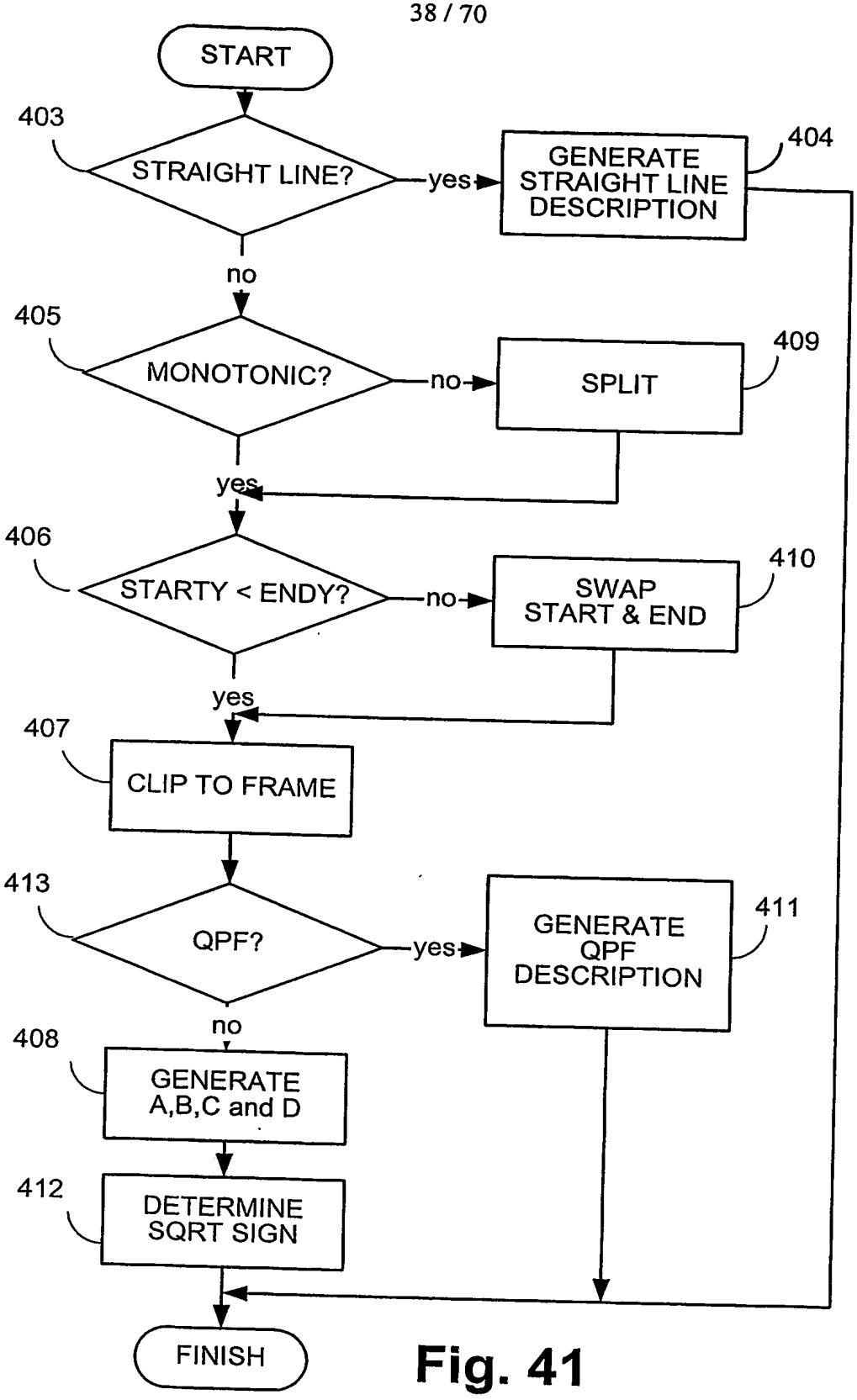

START
403
STRAIGHT LINE?
yes
GENERATE STRAIGHT LINE DESCRIPTION
404
no
405
MONOTONIC?
no
SPLIT
409
yes
406
STARTY < ENDY?
no
SWAP START & END
410
yes
407
CLIP TO FRAME
413
QPF?
yes
GENERATE QPF DESCRIPTION
411
no
408
GENERATE A,B,C and D
412
DETERMINE SQRT SIGN
FINISH

Fig. 41

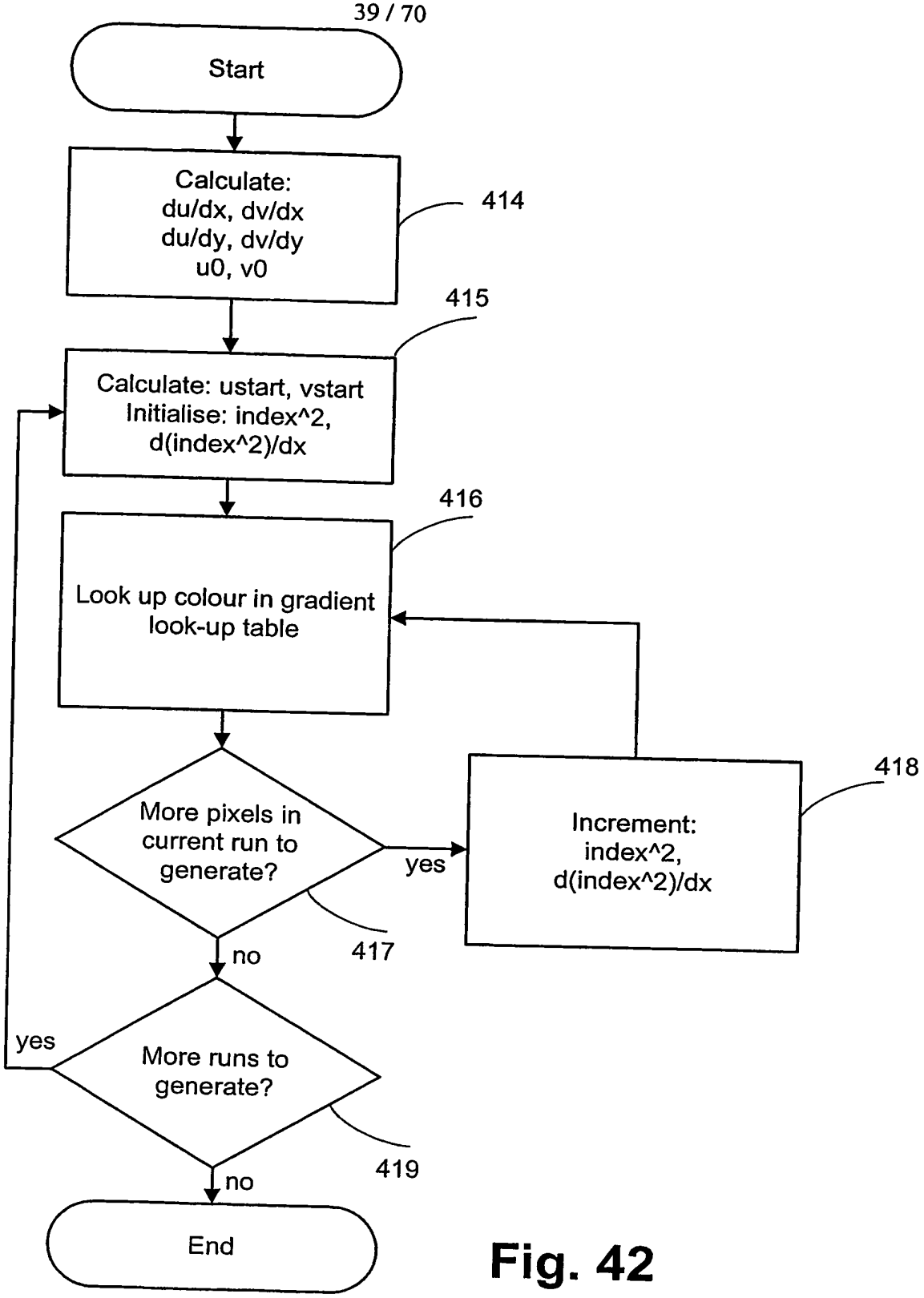

Start
Calculate:
du/dx, dv/dx
du/dy, dv/dy
u0, v0
414
415
Calculate: ustart, vstart
Initialise: index^2,
d(index^2)/dx
416
Look up colour in gradient
look-up table
More pixels in
current run to
generate?
yes
418
Increment:
index^2,
d(index^2)/dx
417
no
yes
More runs to
generate?
419
no
End

Fig. 42

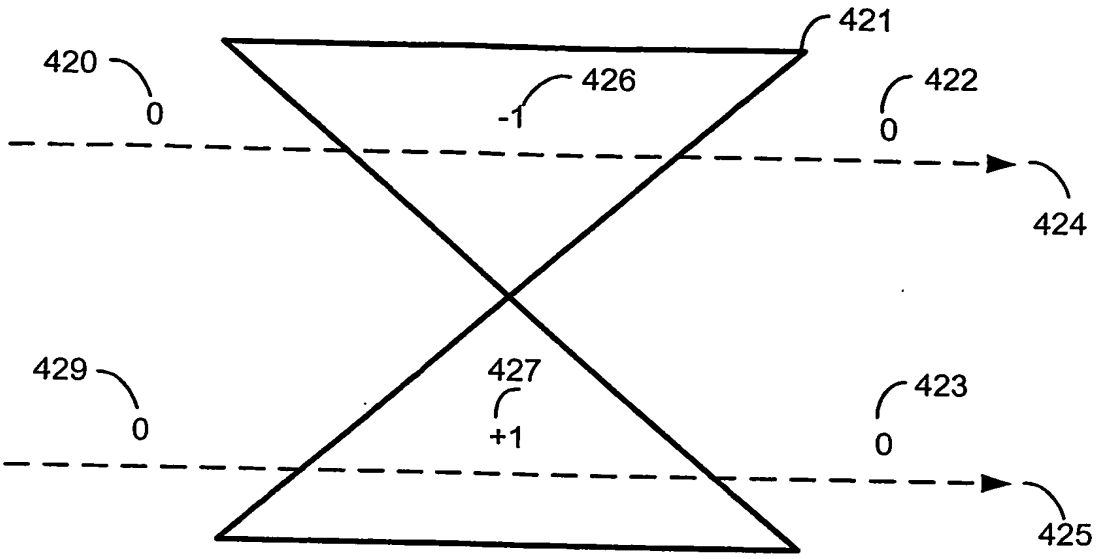


Fig. 43(a)

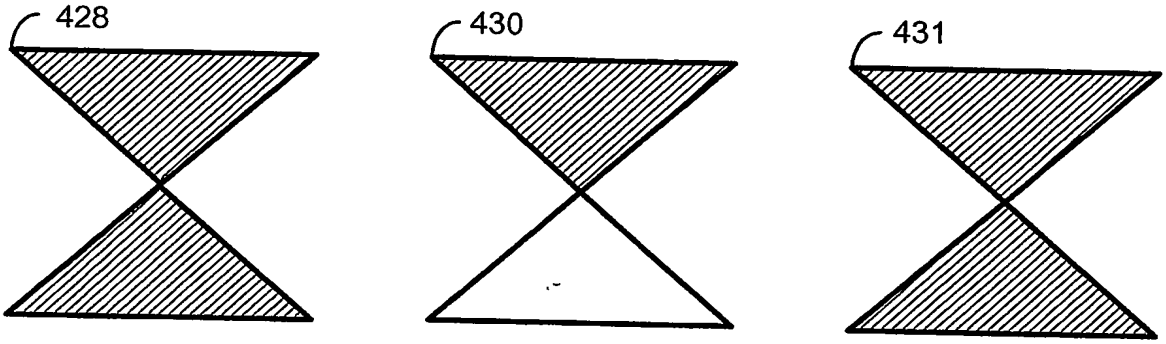


Fig. 43(b)

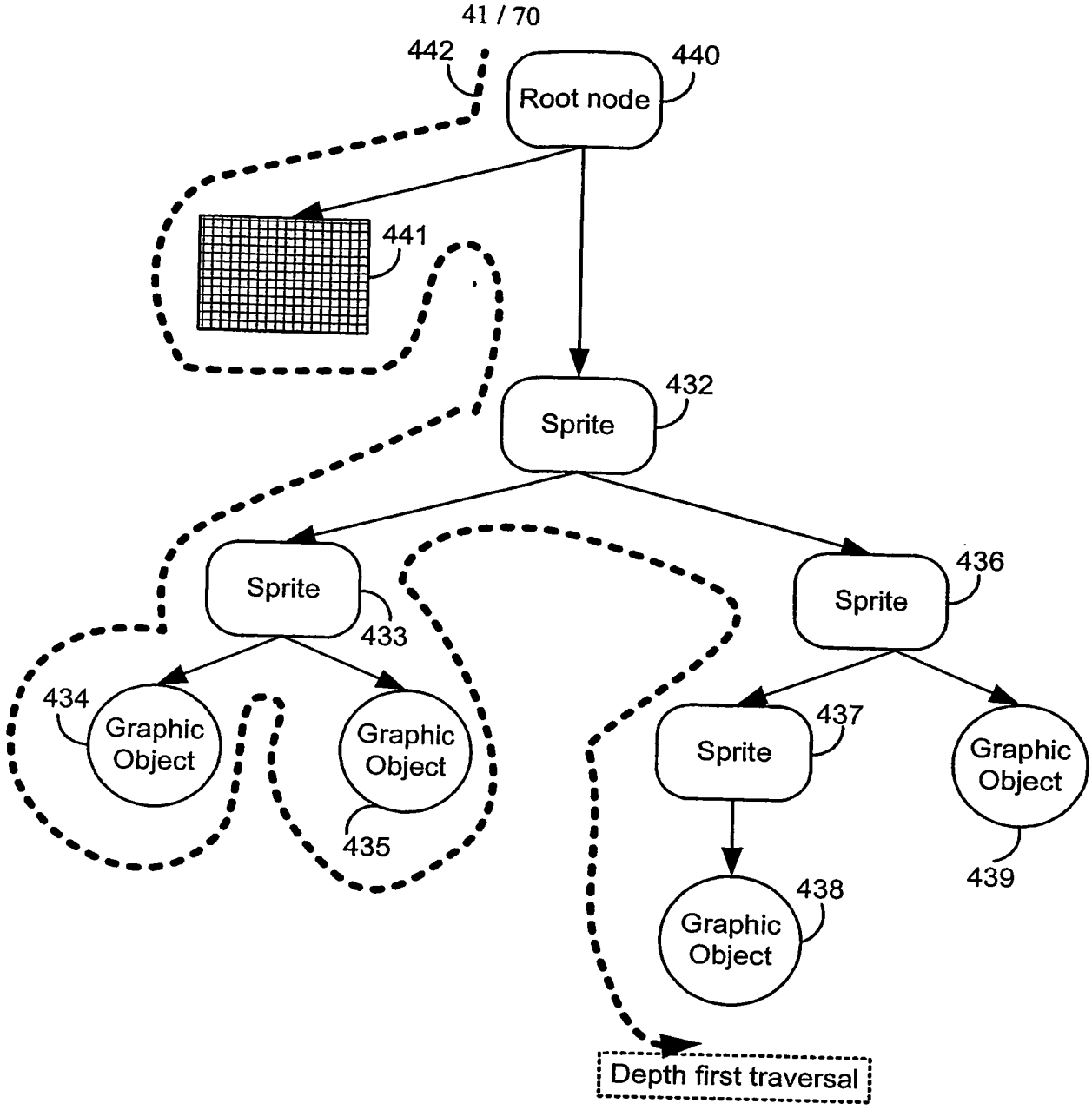

442
Root node
440
441
Sprite
432
Sprite
433
Sprite
436
434
Graphic Object
Graphic Object
435
Sprite
437
Graphic Object
439
438
Graphic Object
Depth first traversal

Fig. 44

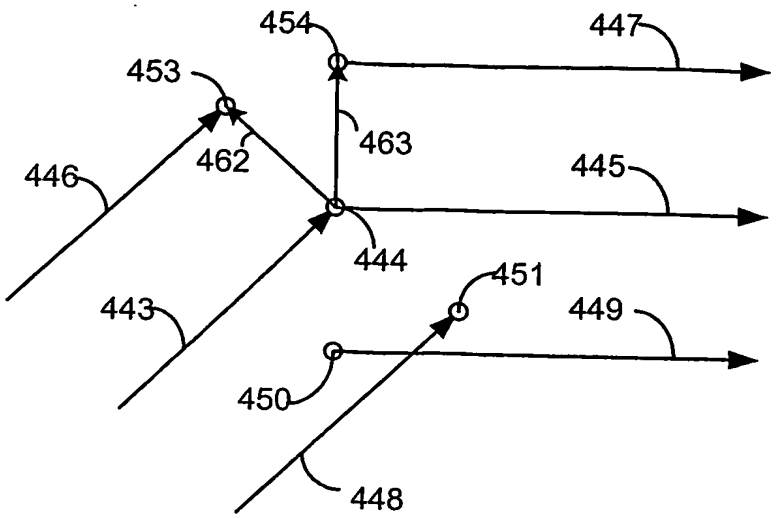


Fig. 45(a)

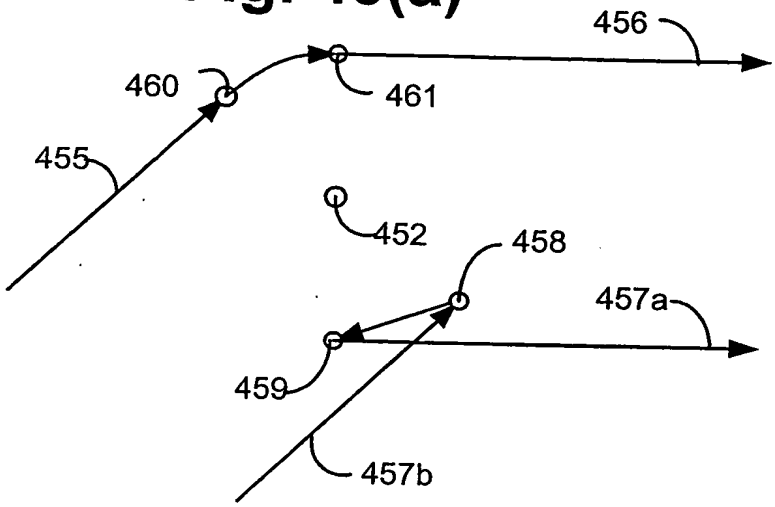


Fig. 45(b)

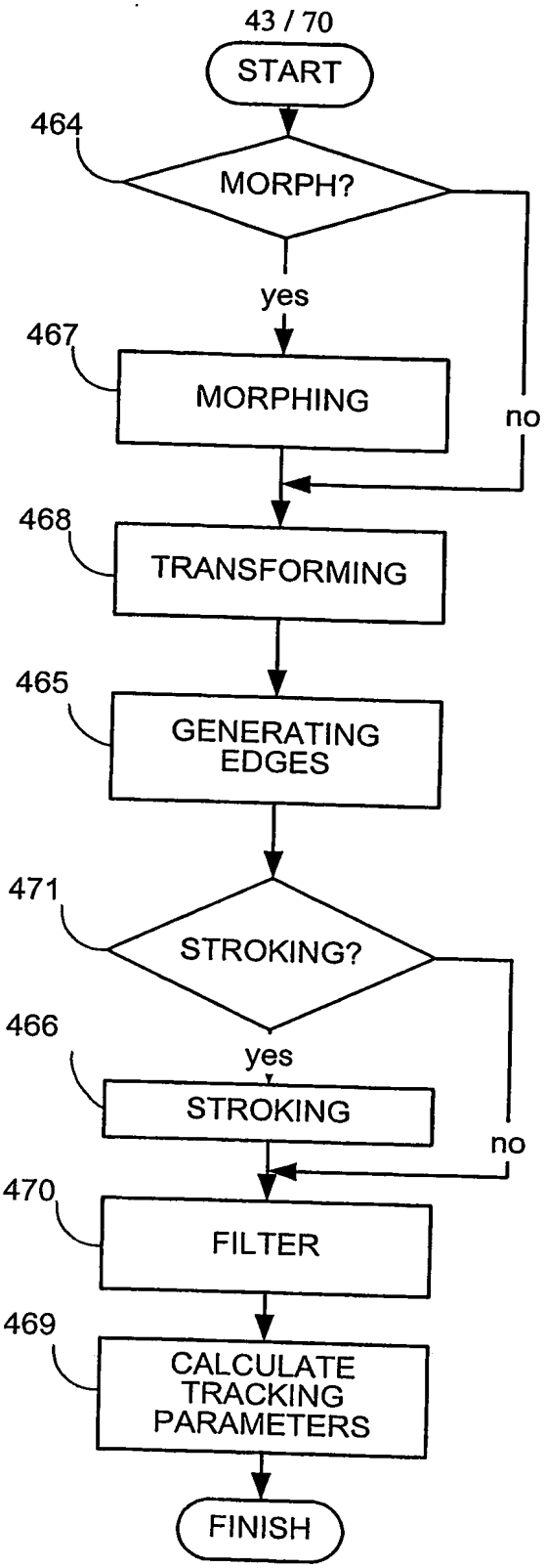

START
464
MORPH?
yes
no
467
MORPHING
468
TRANSFORMING
465
GENERATING EDGES
471
STROKING?
yes
no
466
STROKING
470
FILTER
469
CALCULATE TRACKING PARAMETERS
FINISH

Fig. 46

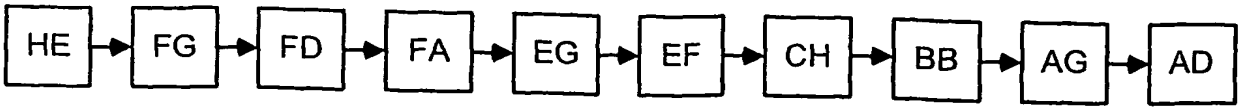


Fig. 47(a)

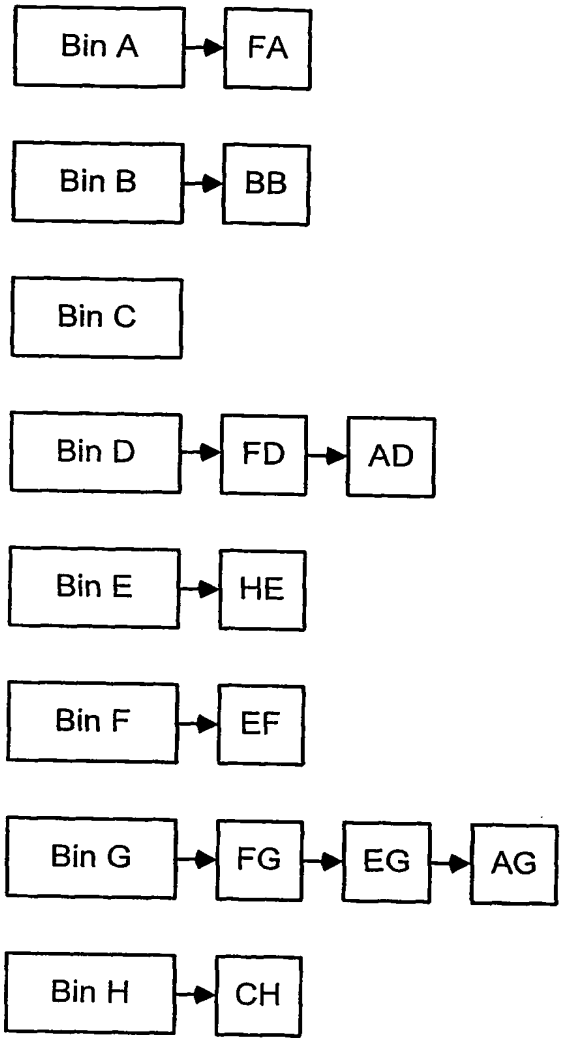


Fig. 47(b)

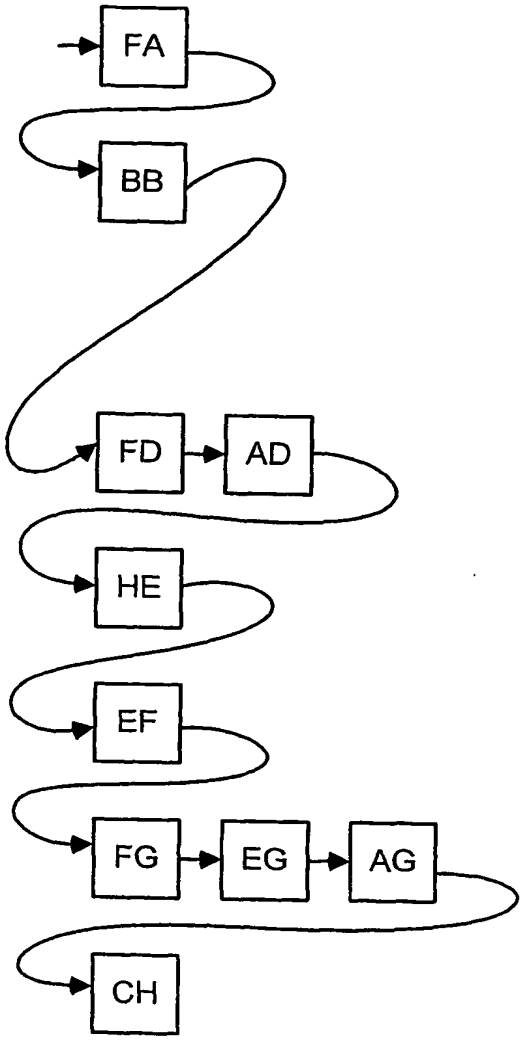


Fig. 47(c)

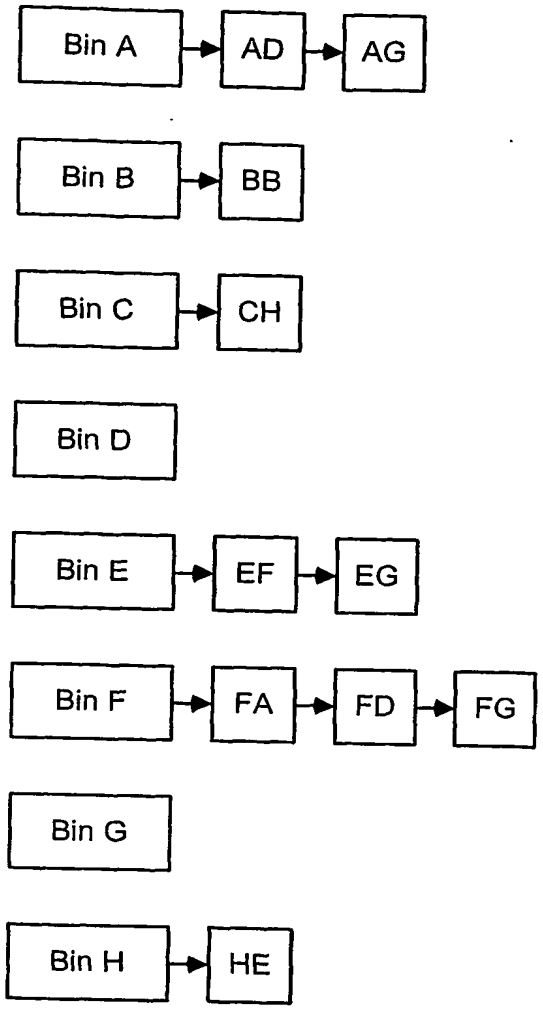


Fig. 47(d)

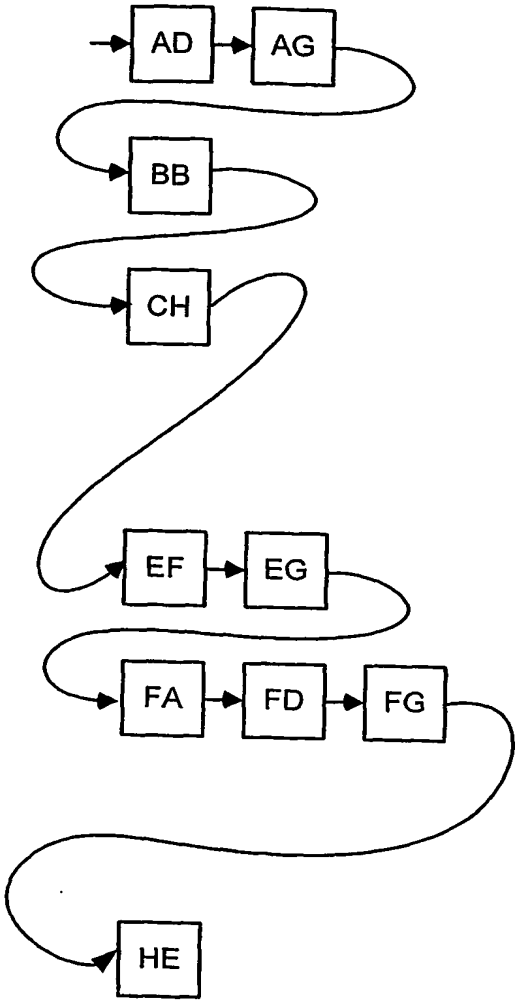


Fig. 47(e)

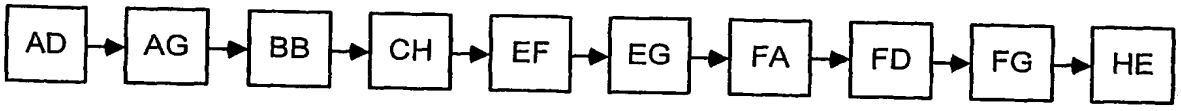


Fig. 47(f)

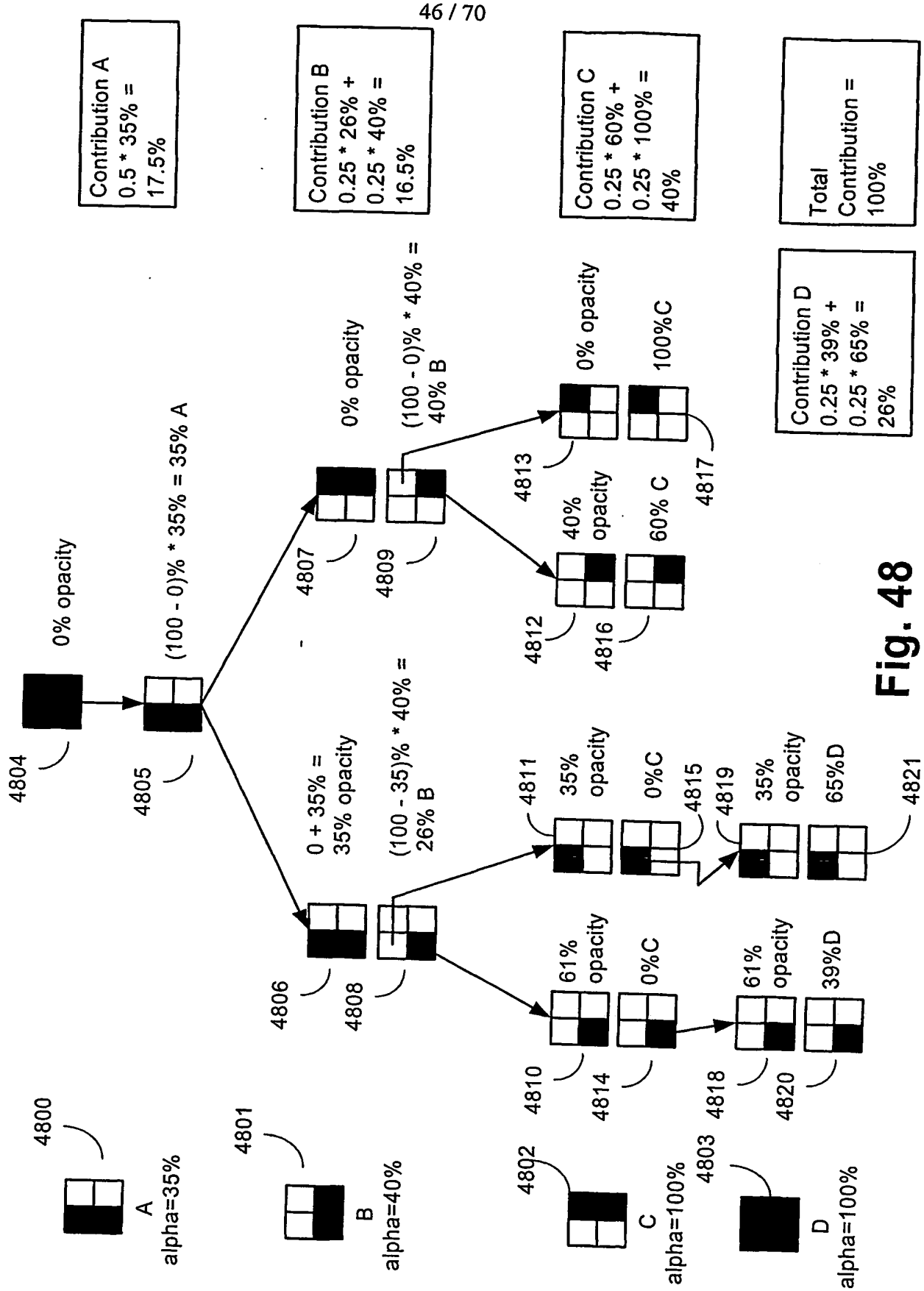

Contribution A
0.5 * 35% =
17.5%
Contribution B
0.25 * 26% +
0.25 * 40% =
16.5%
Contribution C
0.25 * 60% +
0.25 * 100% =
40%
Contribution D
0.25 * 39% +
0.25 * 65% =
26%
Total
Contribution =
100%
4804
0% opacity
4805
(100 - 0)% * 35% = 35% A
4806
0 + 35% =
35% opacity
4807
0% opacity
4808
(100 - 35)% * 40% =
26% B
4809
(100 - 0)% * 40% =
40% B
4810
61%
opacity
4811
35%
opacity
4812
40%
opacity
4813
0% opacity
4814
0%C
4815
0%C
4816
60% C
4817
100%C
4818
61%
opacity
4819
35%
opacity
4820
39%D
4821
65%D
4800
A
alpha=35%
4801
B
alpha=40%
4802
C
alpha=100%
4803
D
alpha=100%

Fig. 48

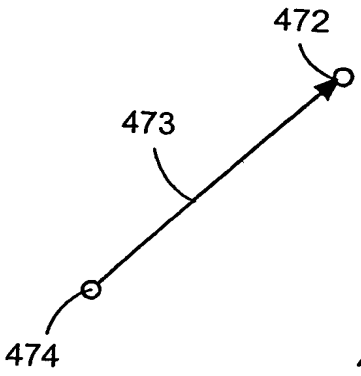


Fig. 49(a)

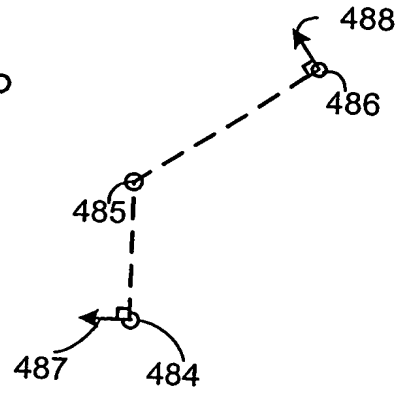


Fig. 49(d)

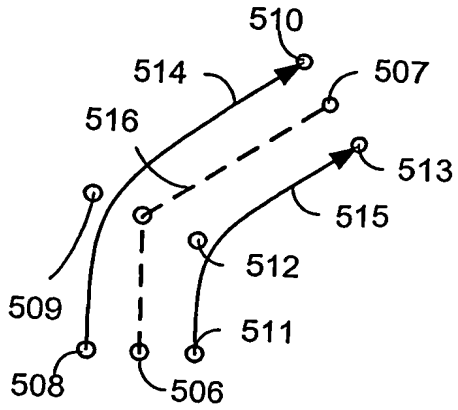


Fig. 49(g)

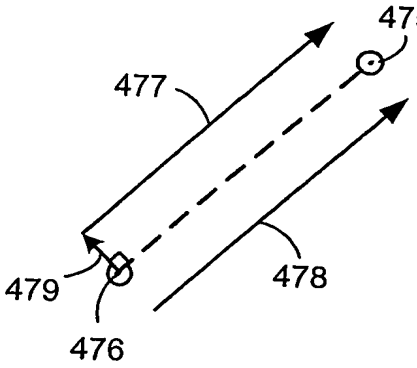


Fig. 49(b)

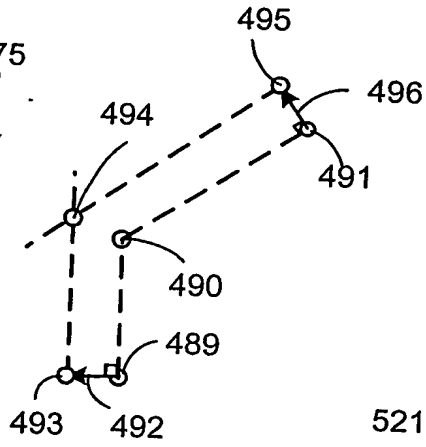


Fig. 49(e)

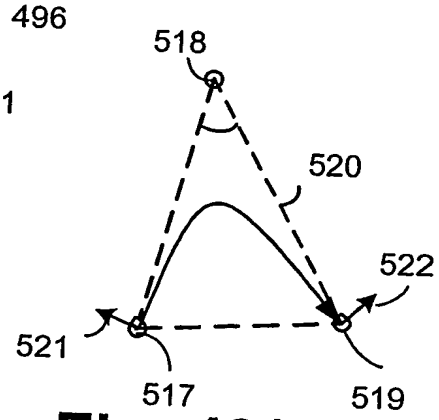


Fig. 49(h)

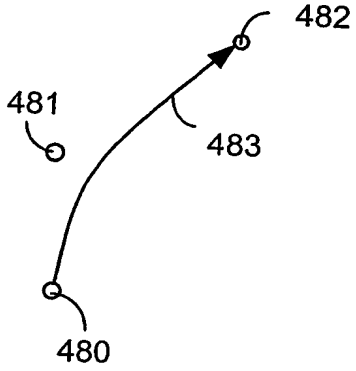


Fig. 49(c)

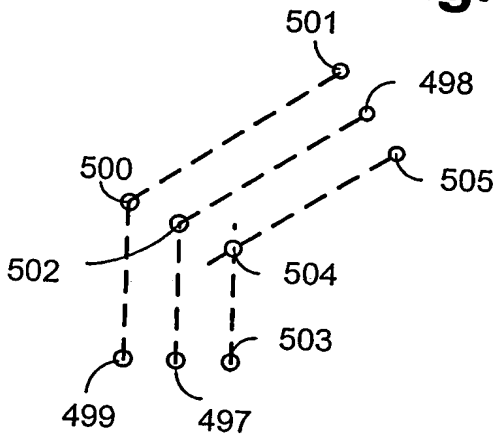


Fig. 49(f)

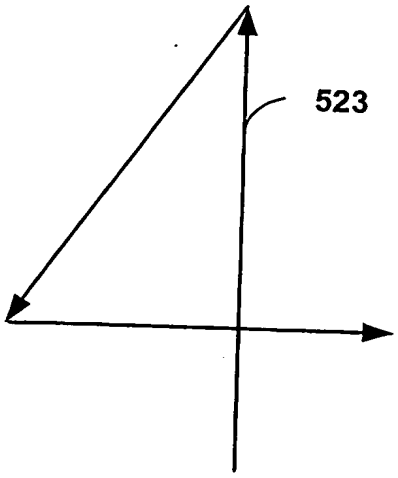


Fig. 50(a)

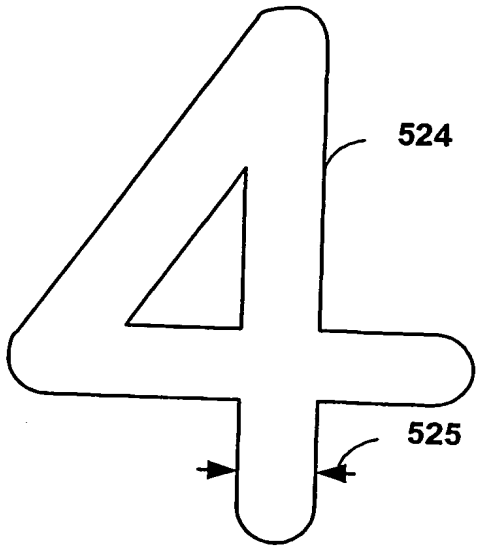


Fig. 50(b)

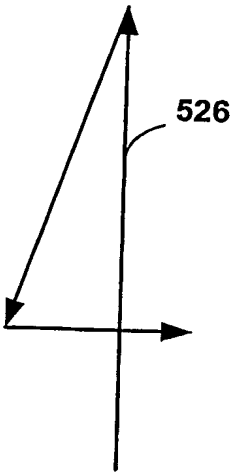


Fig. 50(c)

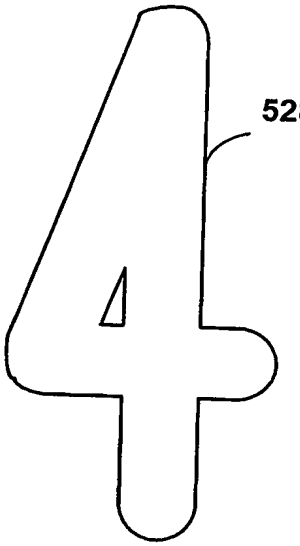


Fig. 50(d)

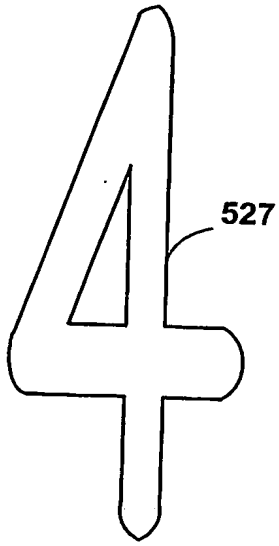


Fig. 50(e)

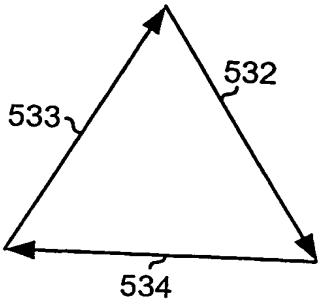
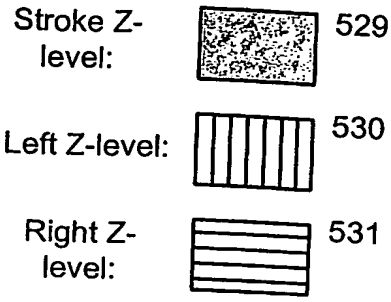


Fig 51(a)



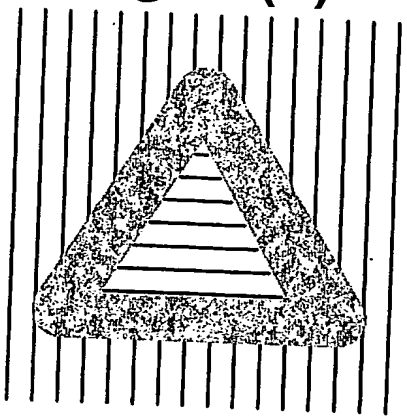


Fig 51(b)

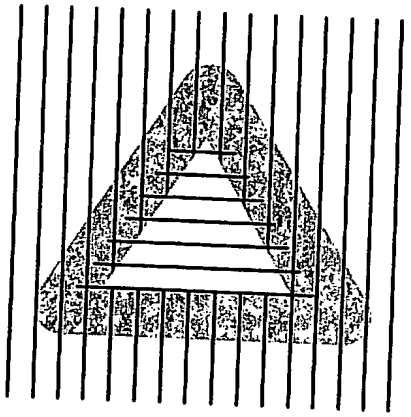


Fig 51(c)

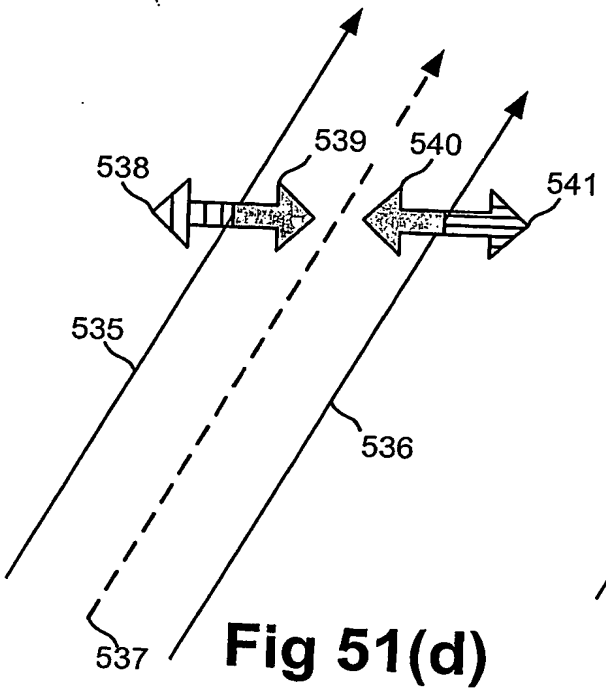


Fig 51(d)

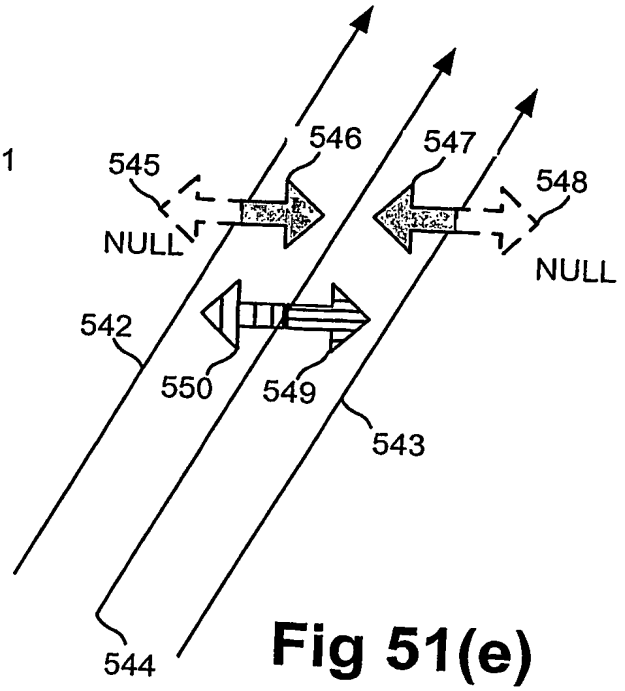


Fig 51(e)

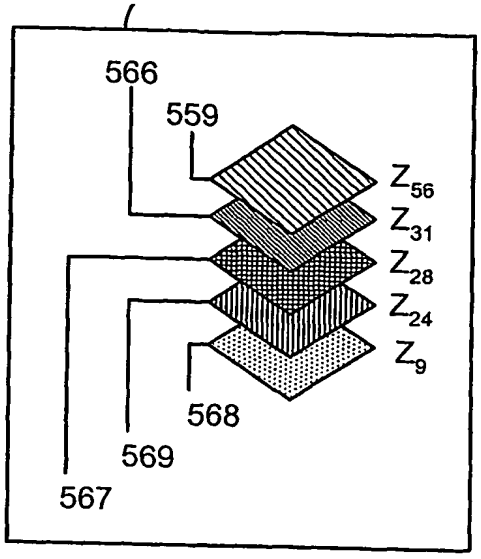


Fig. 52(a)

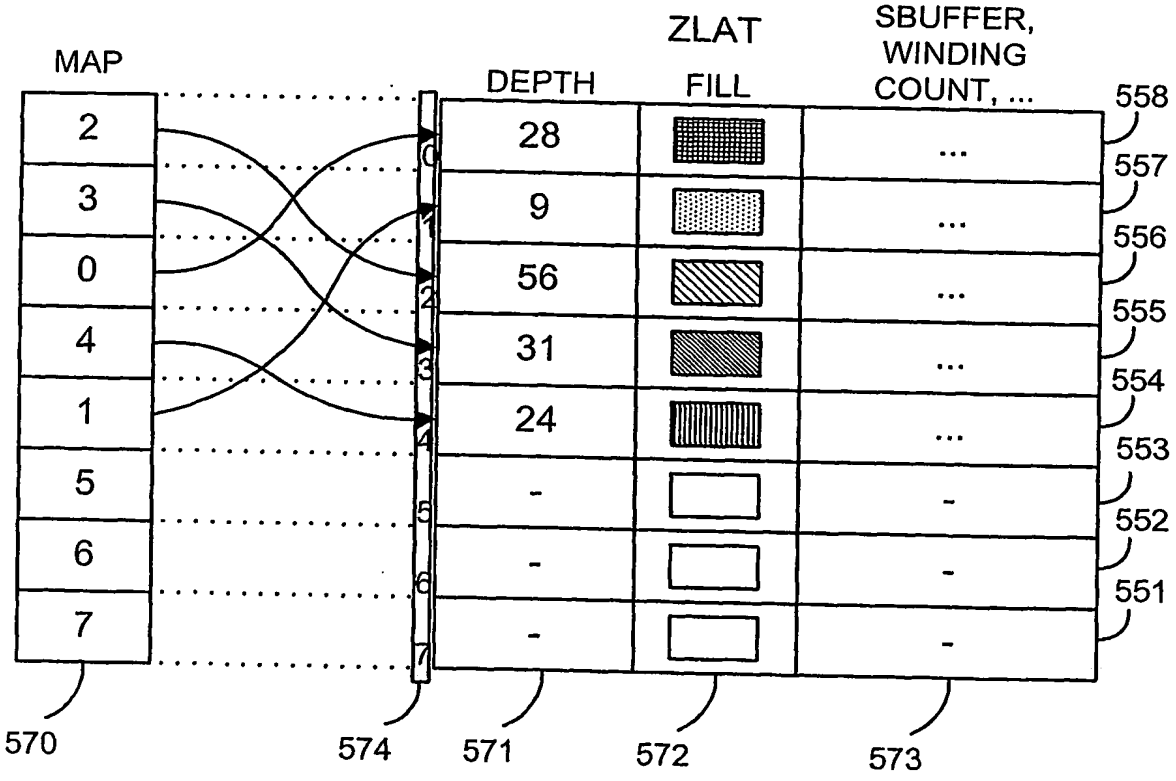


Fig. 52(b)

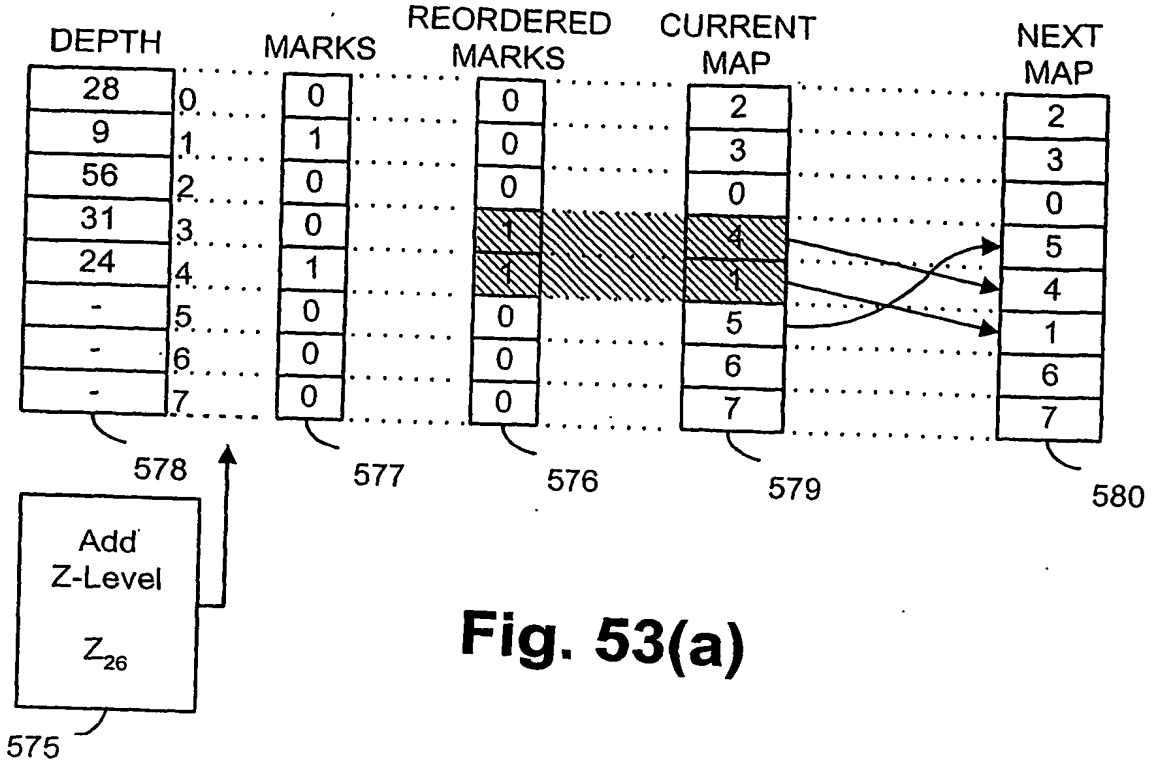


Fig. 53(a)

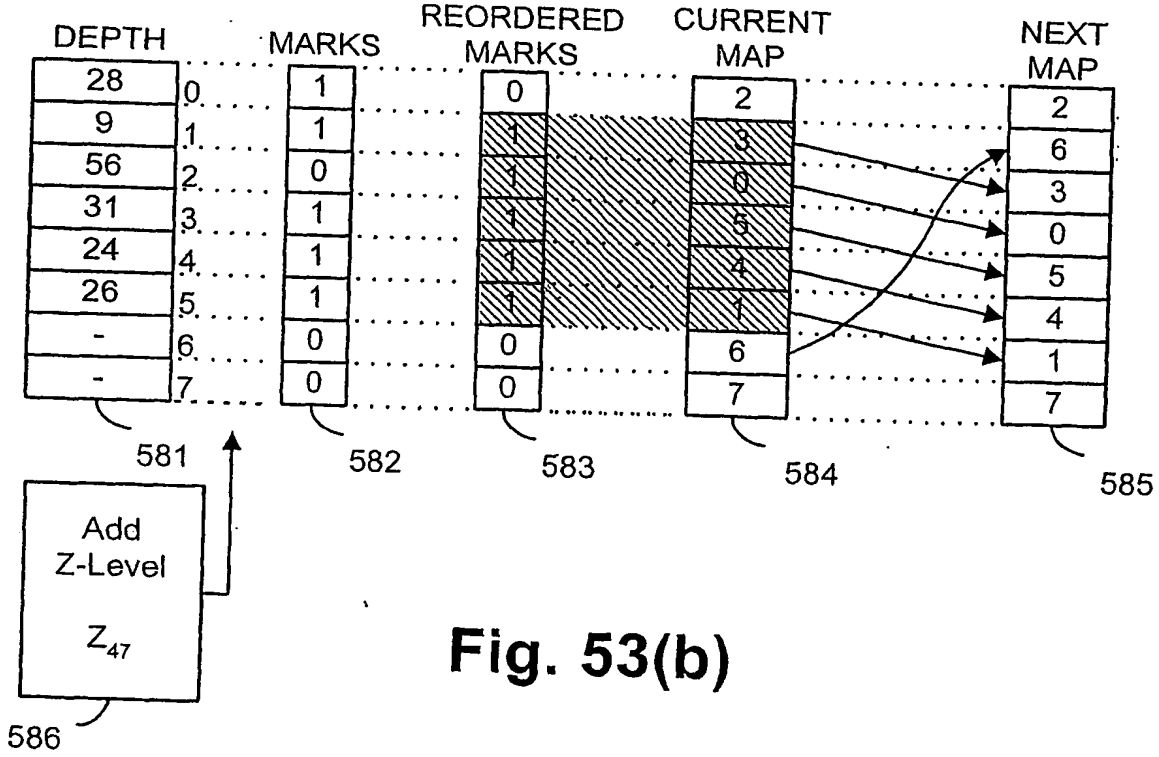


Fig. 53(b)

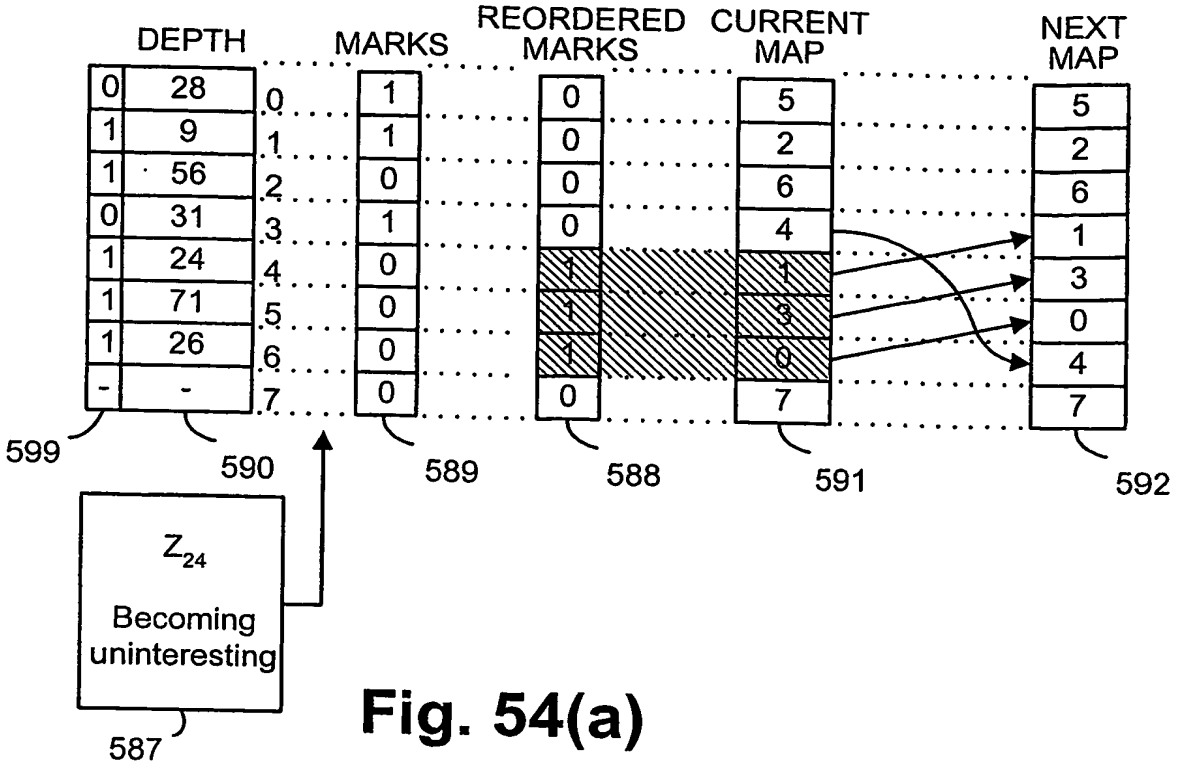


Fig. 54(a)

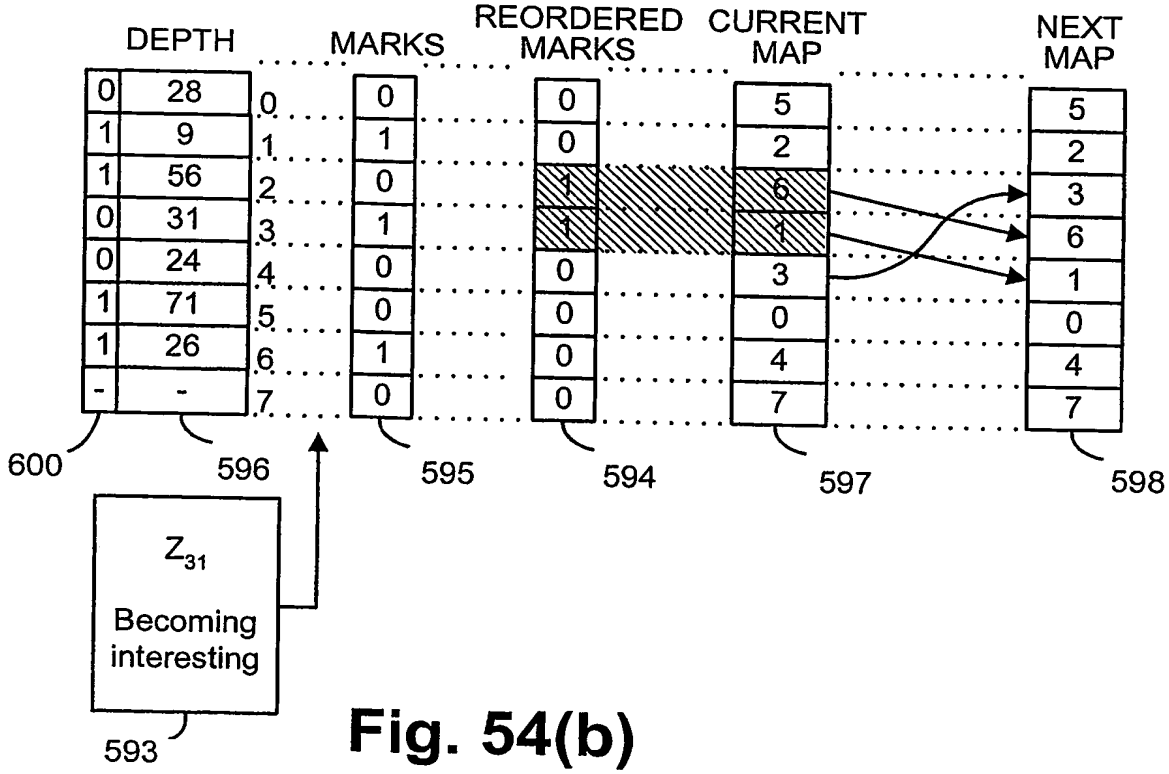


Fig. 54(b)

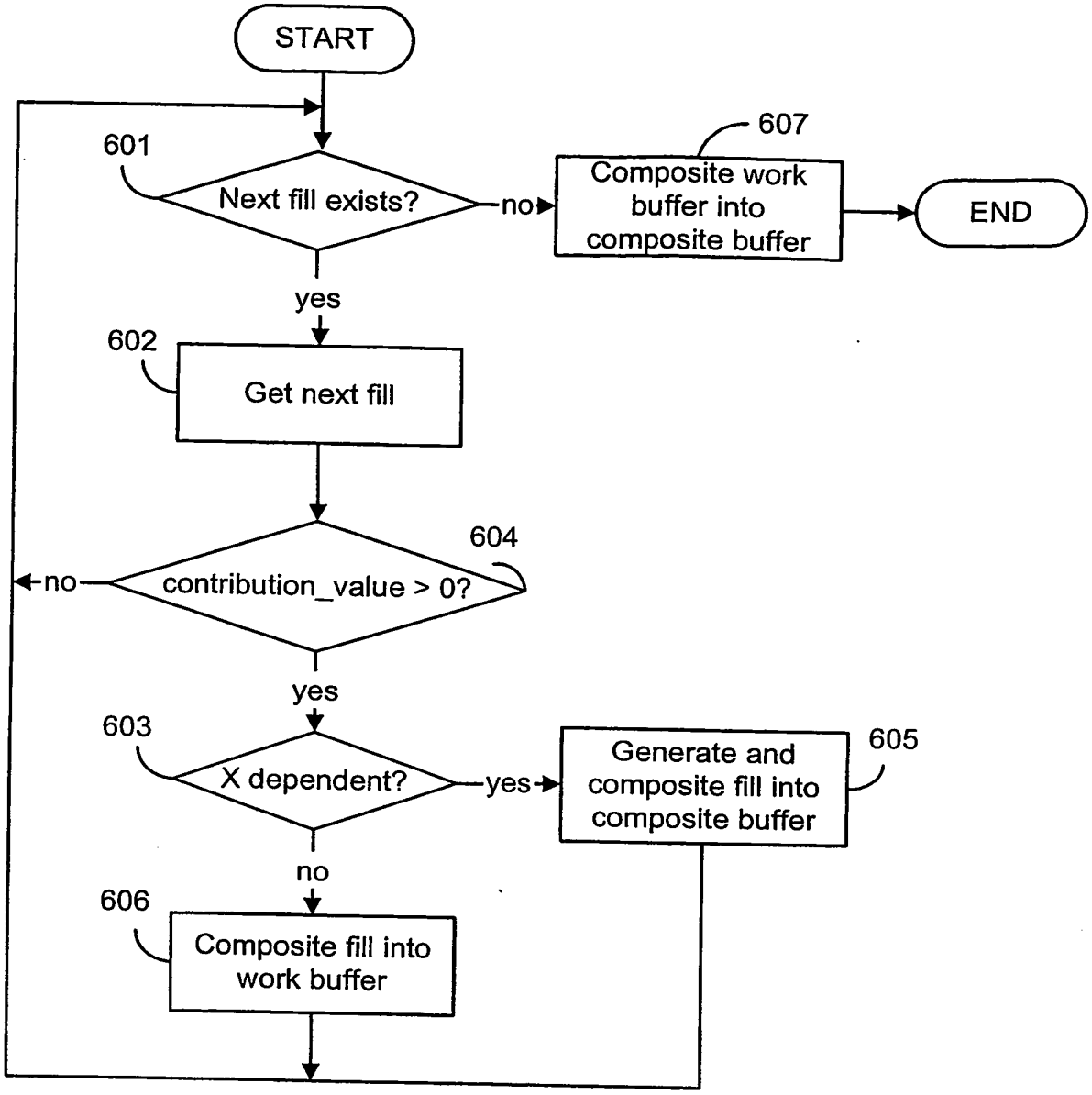

START
601
Next fill exists?
no
607
Composite work buffer into composite buffer
END
yes
602
Get next fill
604
contribution_value > 0?
no
yes
603
X dependent?
yes
605
Generate and composite fill into composite buffer
no
606
Composite fill into work buffer

Fig. 55

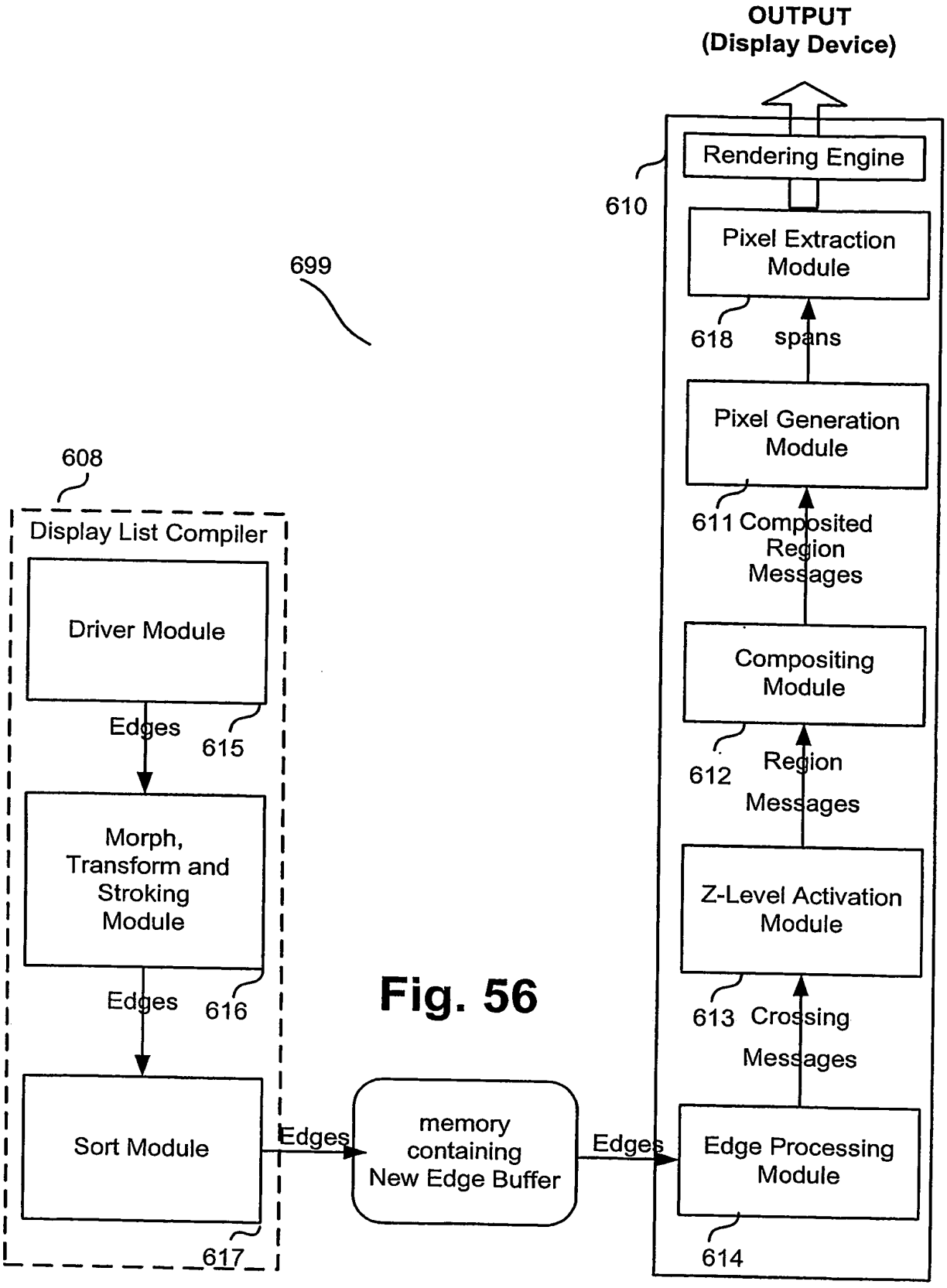

OUTPUT
(Display Device)
Rendering Engine
610
Pixel Extraction Module
699
618
spans
Pixel Generation Module
608
611 Composited Region Messages
Display List Compiler
Compositing Module
Driver Module
Edges
615
612
Region Messages
Morph, Transform and Stroking Module
Z-Level Activation Module
Edges
616
613 Crossing Messages
Sort Module
Edges
memory containing New Edge Buffer
Edges
Edge Processing Module
617
614

Fig. 56

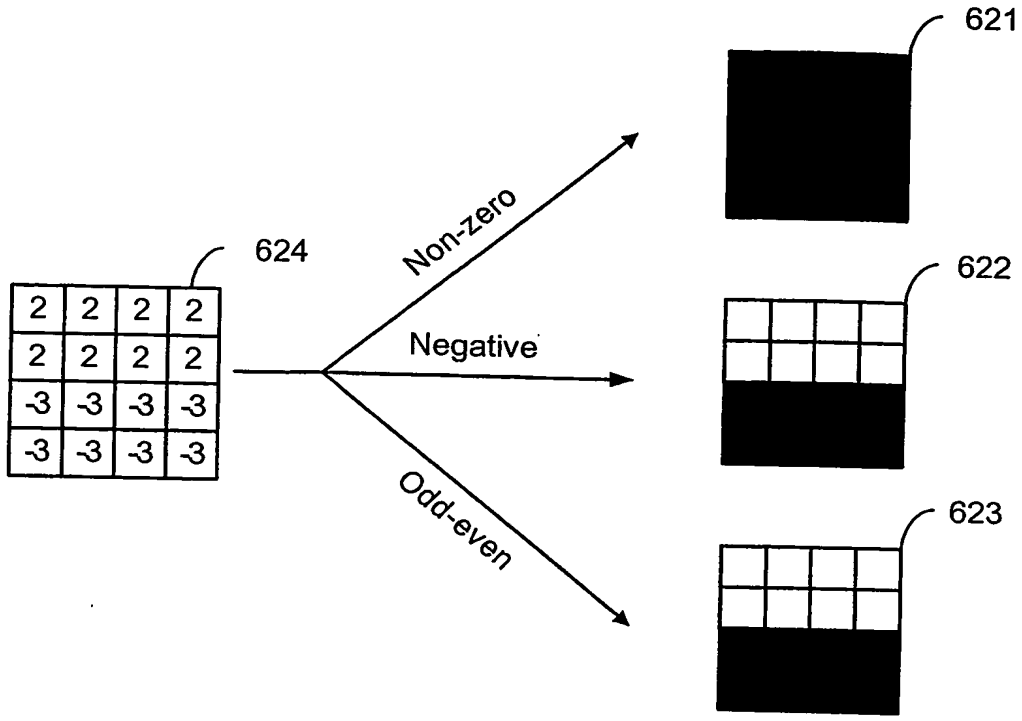


Fig. 57(a)

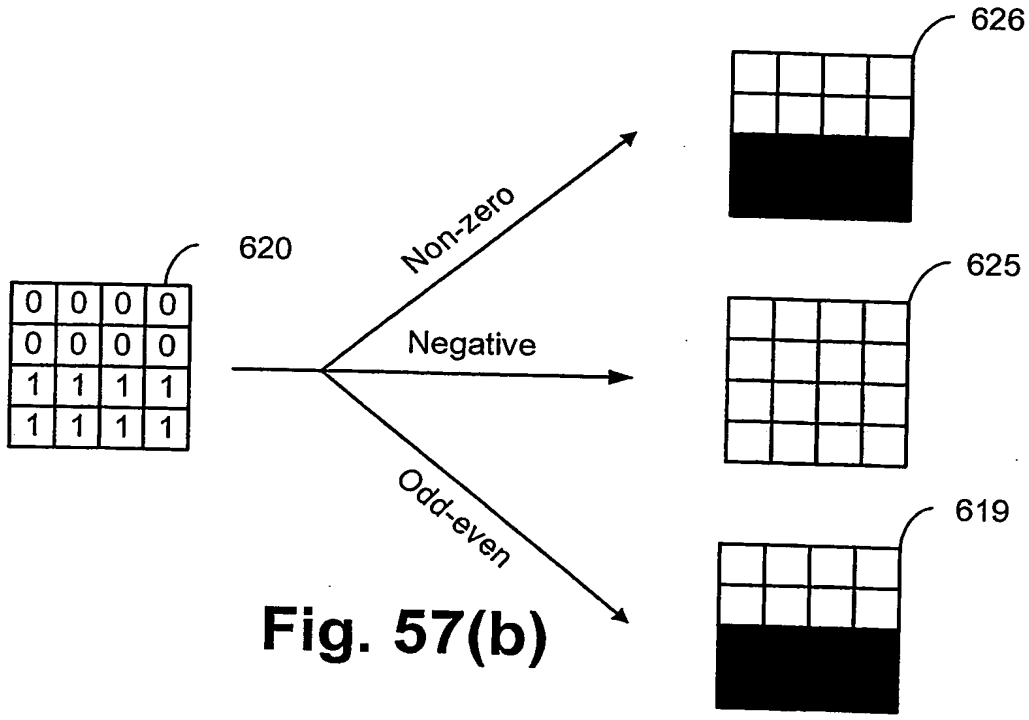


Fig. 57(b)

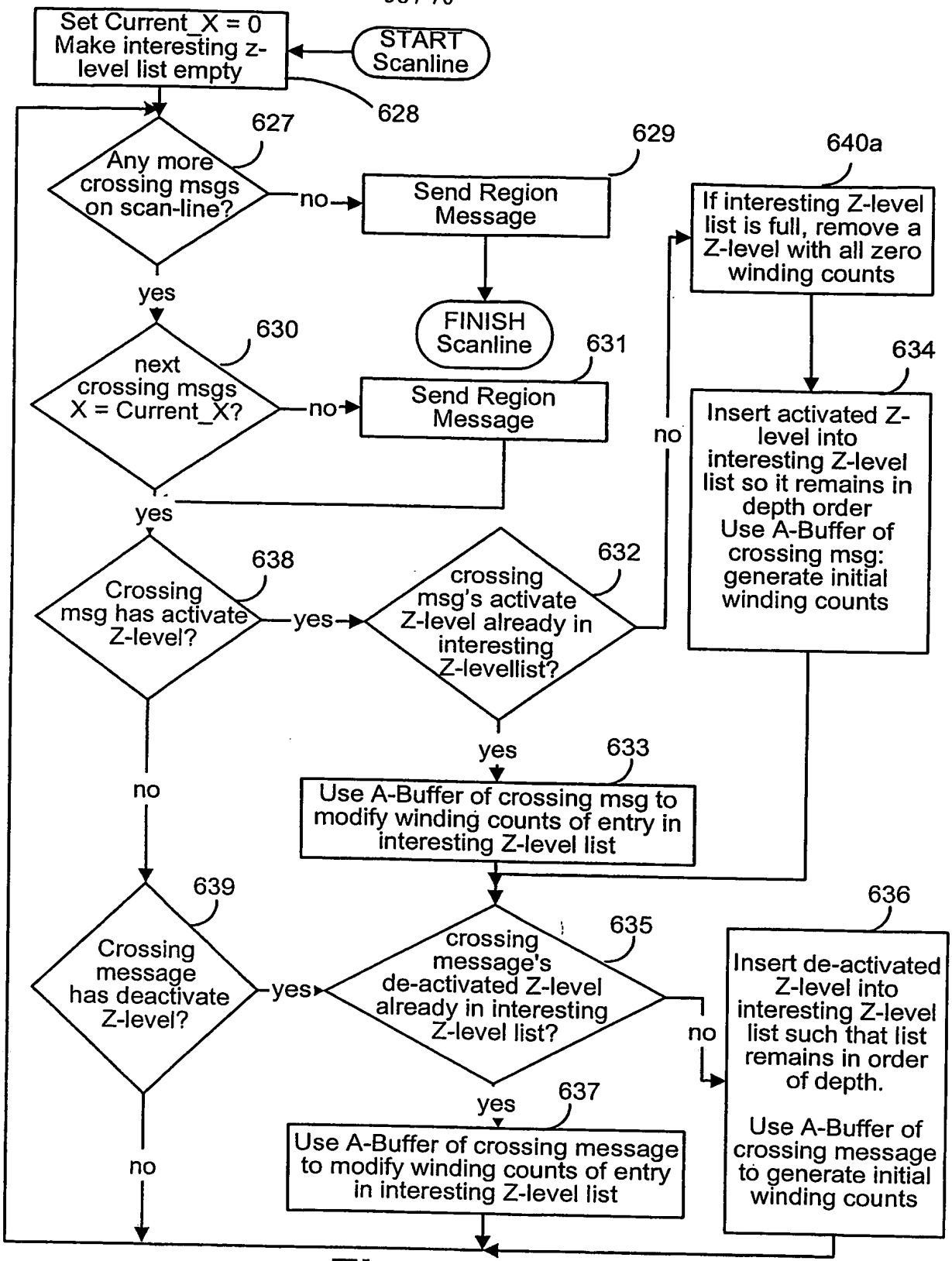

Set Current_X = 0
Make interesting z-level list empty
START Scanline
628
627
Any more crossing msgs on scan-line?
no
629
Send Region Message
FINISH Scanline
yes
630
next crossing msgs X = Current_X?
no
631
Send Region Message
yes
638
Crossing msg has activate Z-level?
yes
632
crossing msg's activate Z-level already in interesting Z-levellist?
no
640a
If interesting Z-level list is full, remove a Z-level with all zero winding counts
634
Insert activated Z-level into interesting Z-level list so it remains in depth order
Use A-Buffer of crossing msg: generate initial winding counts
yes
633
Use A-Buffer of crossing msg to modify winding counts of entry in interesting Z-level list
no
639
Crossing message has deactivate Z-level?
yes
635
crossing message's de-activated Z-level already in interesting Z-level list?
no
636
Insert de-activated Z-level into interesting Z-level list such that list remains in order of depth.
Use A-Buffer of crossing message to generate initial winding counts
yes
637
Use A-Buffer of crossing message to modify winding counts of entry in interesting Z-level list
no

Fig. 58

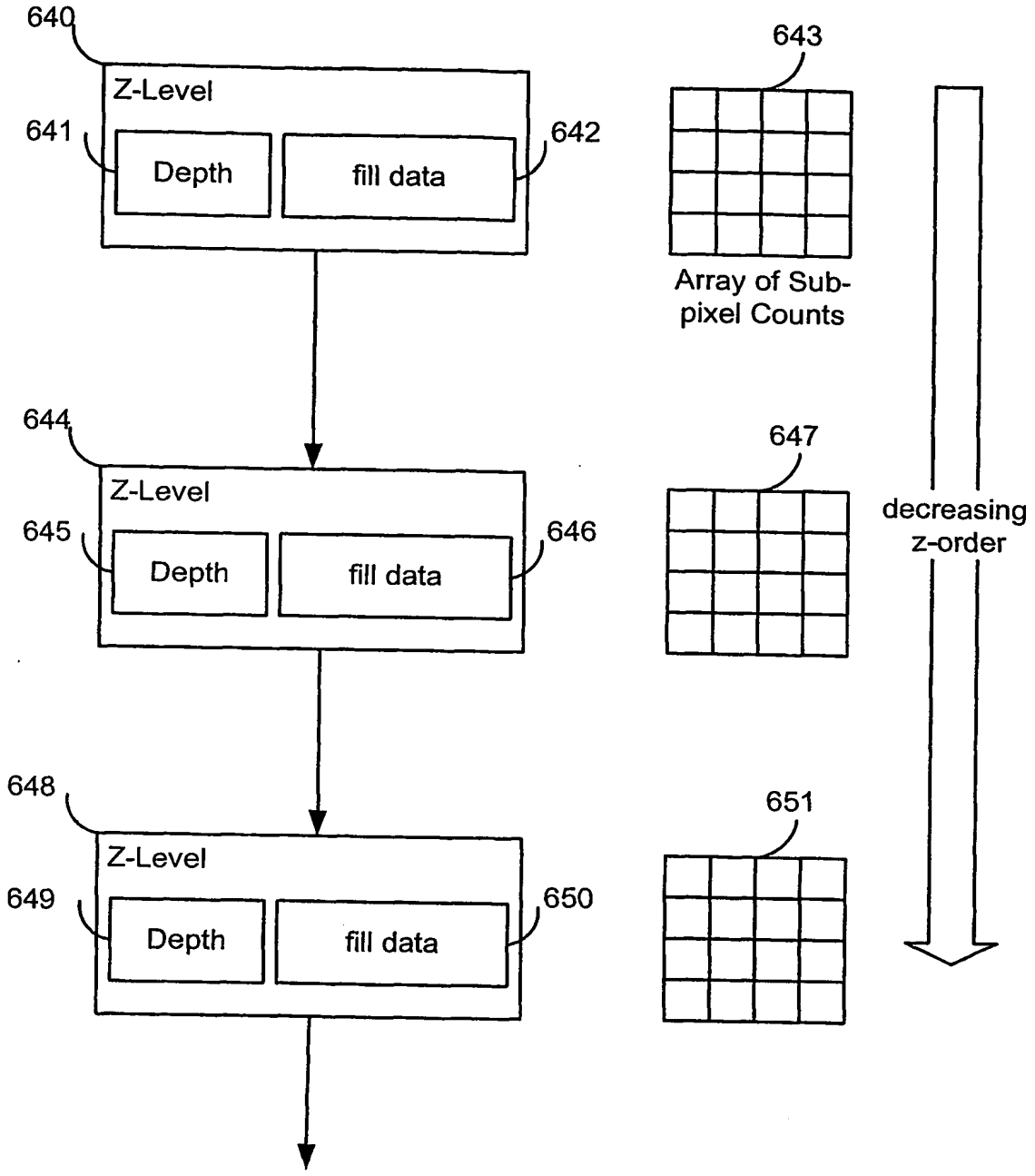

640
Z-Level
641
Depth
642
fill data
643
Array of Sub-pixel Counts
644
Z-Level
645
Depth
646
fill data
647
decreasing z-order
648
Z-Level
649
Depth
650
fill data
651

Fig. 59

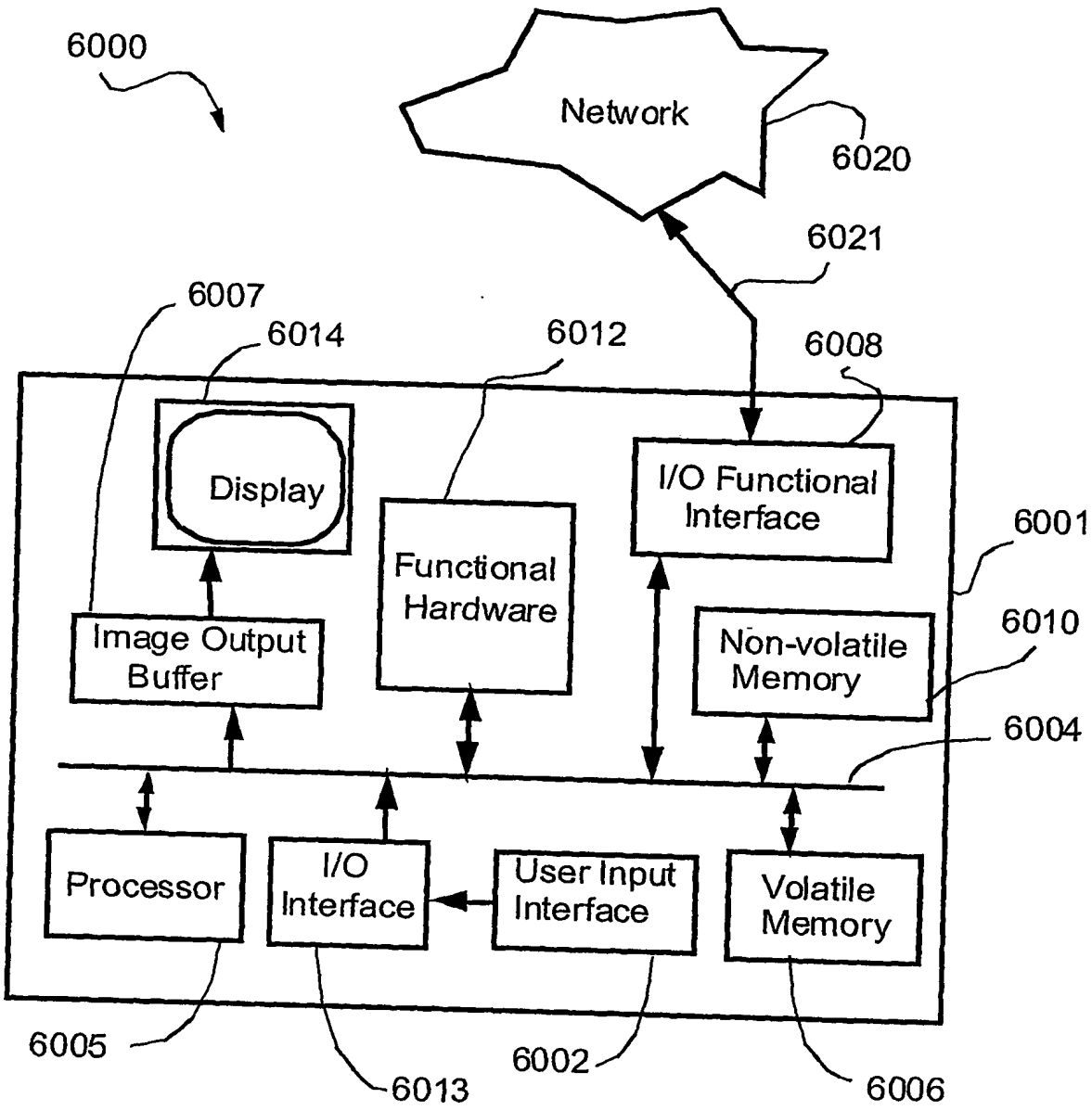

6000
Network
6020
6021
6007
6014
6012
6008
Display
I/O Functional Interface
6001
Functional Hardware
Image Output Buffer
Non-volatile Memory
6010
6004
Processor
I/O Interface
User Input Interface
Volatile Memory
6005
6013
6002
6006

Fig. 60

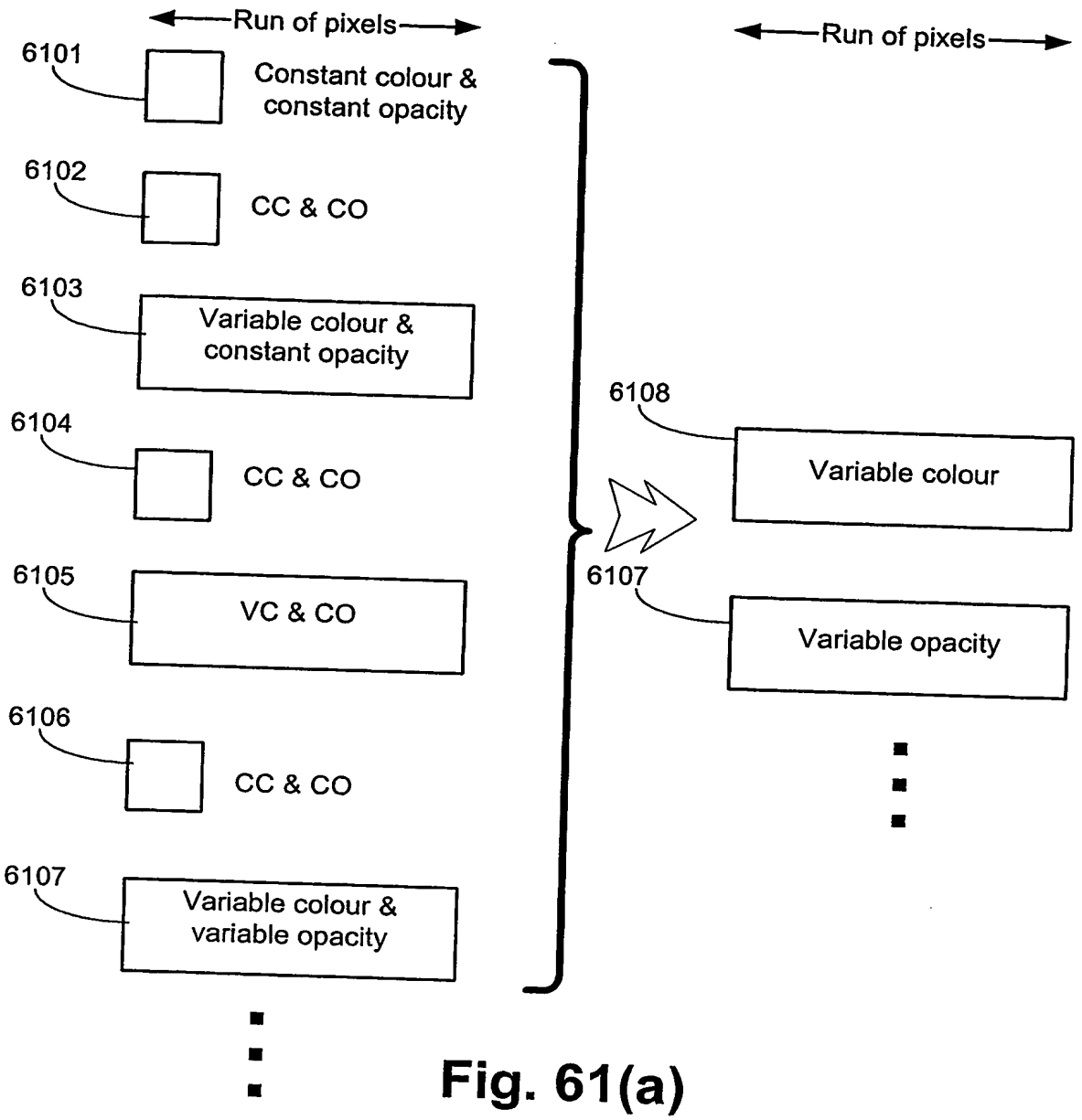

Run of pixels
6101
Constant colour & constant opacity
6102
CC & CO
6103
Variable colour & constant opacity
6104
CC & CO
6105
VC & CO
6106
CC & CO
6107
Variable colour & variable opacity
Run of pixels
6108
Variable colour
6107
Variable opacity

Fig. 61(a)

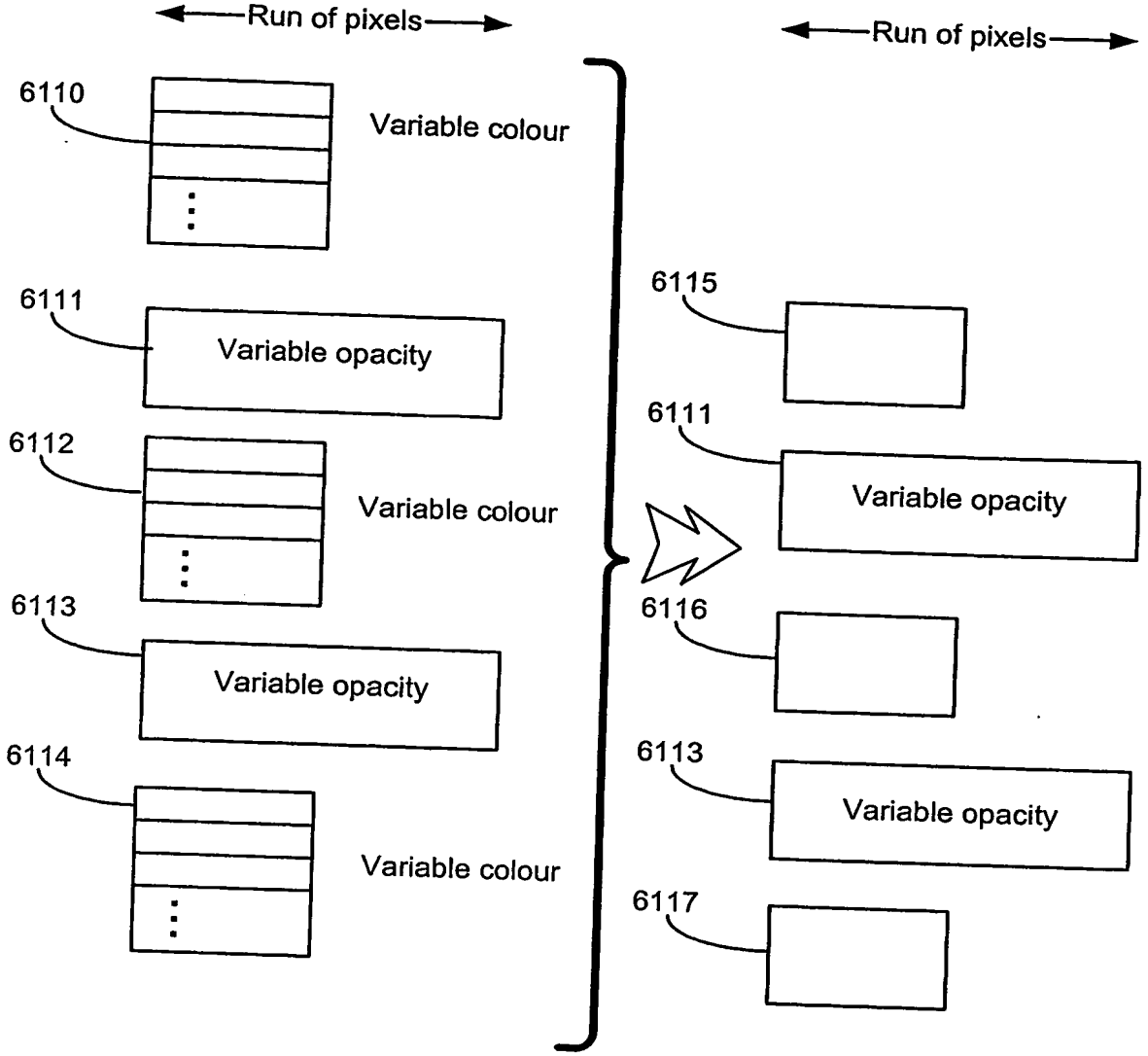

Run of pixels
Run of pixels
6110
Variable colour
6111
Variable opacity
6112
Variable colour
6113
Variable opacity
6114
Variable colour
6115
6111
Variable opacity
6116
6113
Variable opacity
6117

Fig. 61(b)

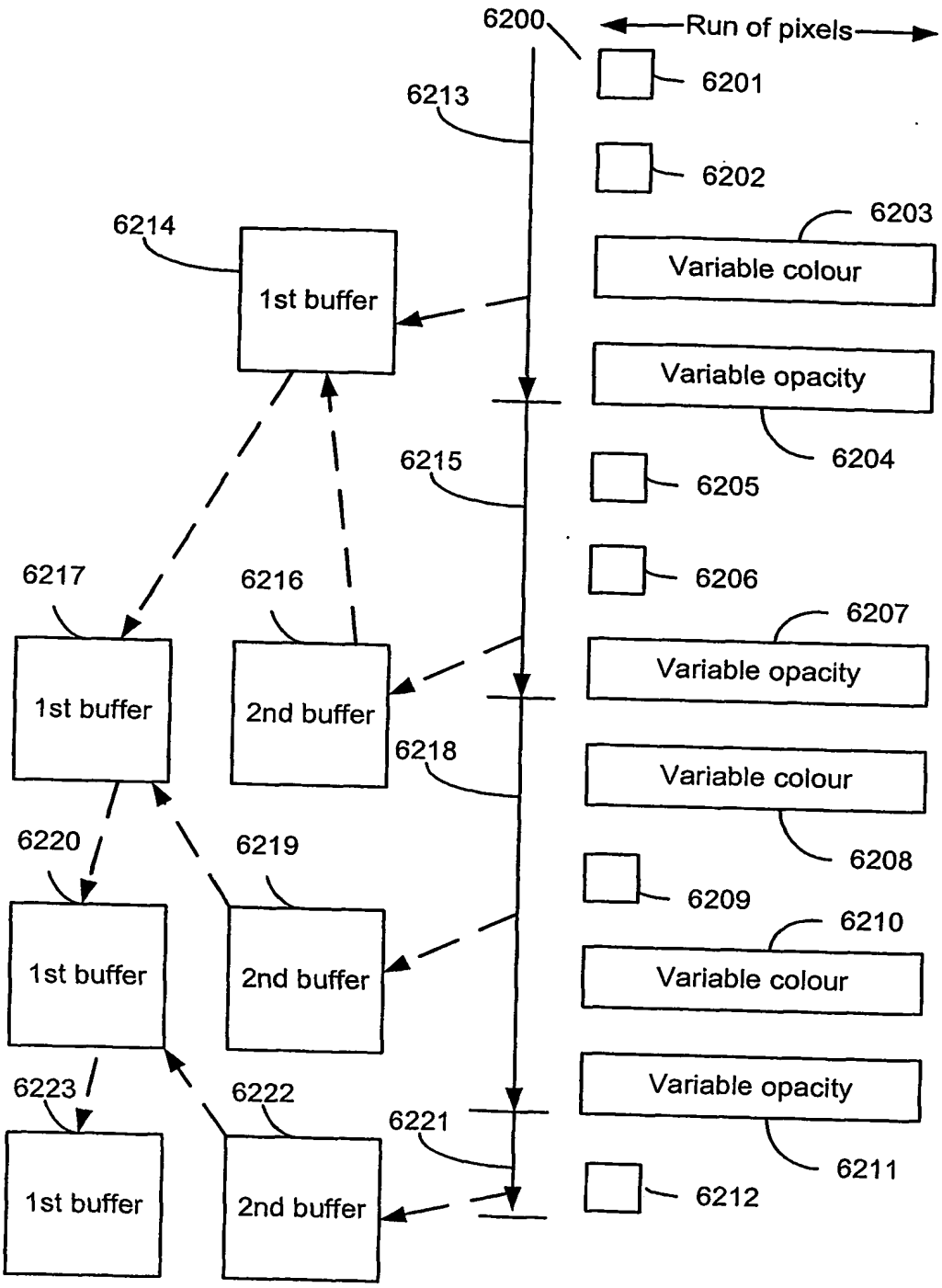

6200
Run of pixels
6201
6213
6202
6214
6203
1st buffer
Variable colour
Variable opacity
6204
6215
6205
6206
6217
6216
6207
1st buffer
2nd buffer
Variable opacity
6218
Variable colour
6208
6220
6219
6209
6210
1st buffer
2nd buffer
Variable colour
Variable opacity
6223
6222
6221
6211
1st buffer
2nd buffer
6212

Fig. 62

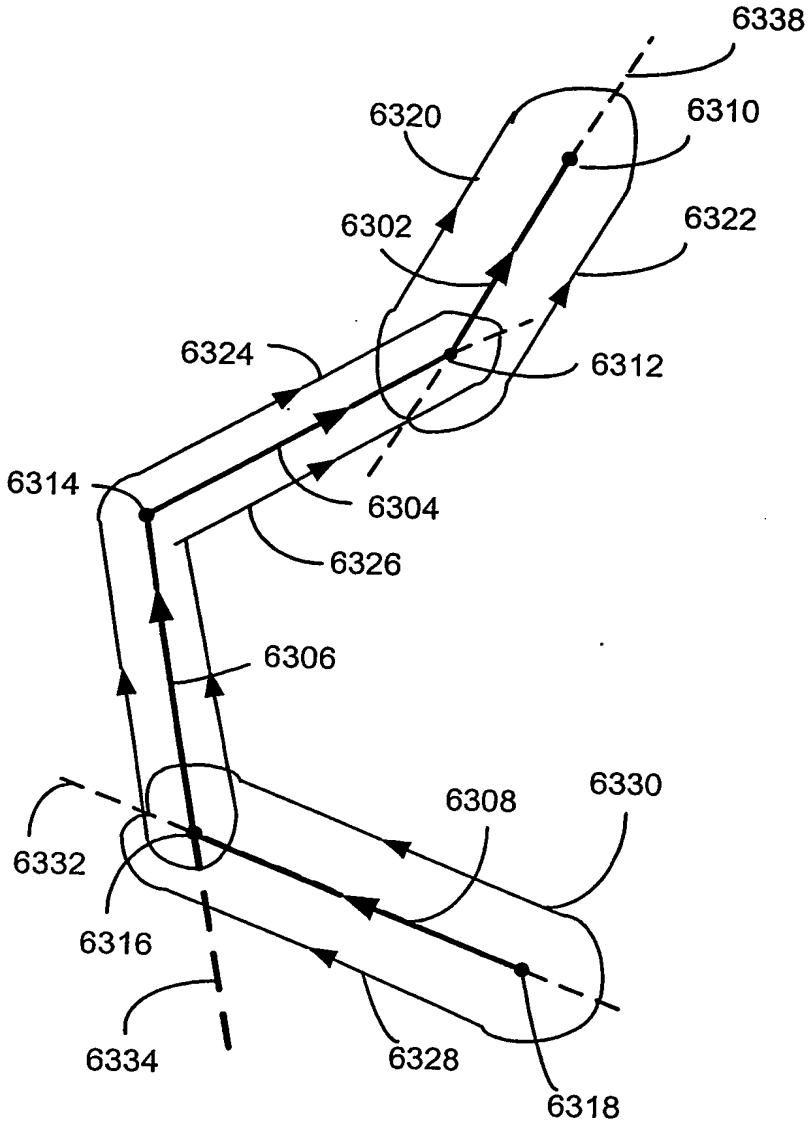

6338
6320
6310
6302
6322
6324
6312
6314
6304
6326
6306
6308
6330
6332
6316
6334
6328
6318

Fig. 63

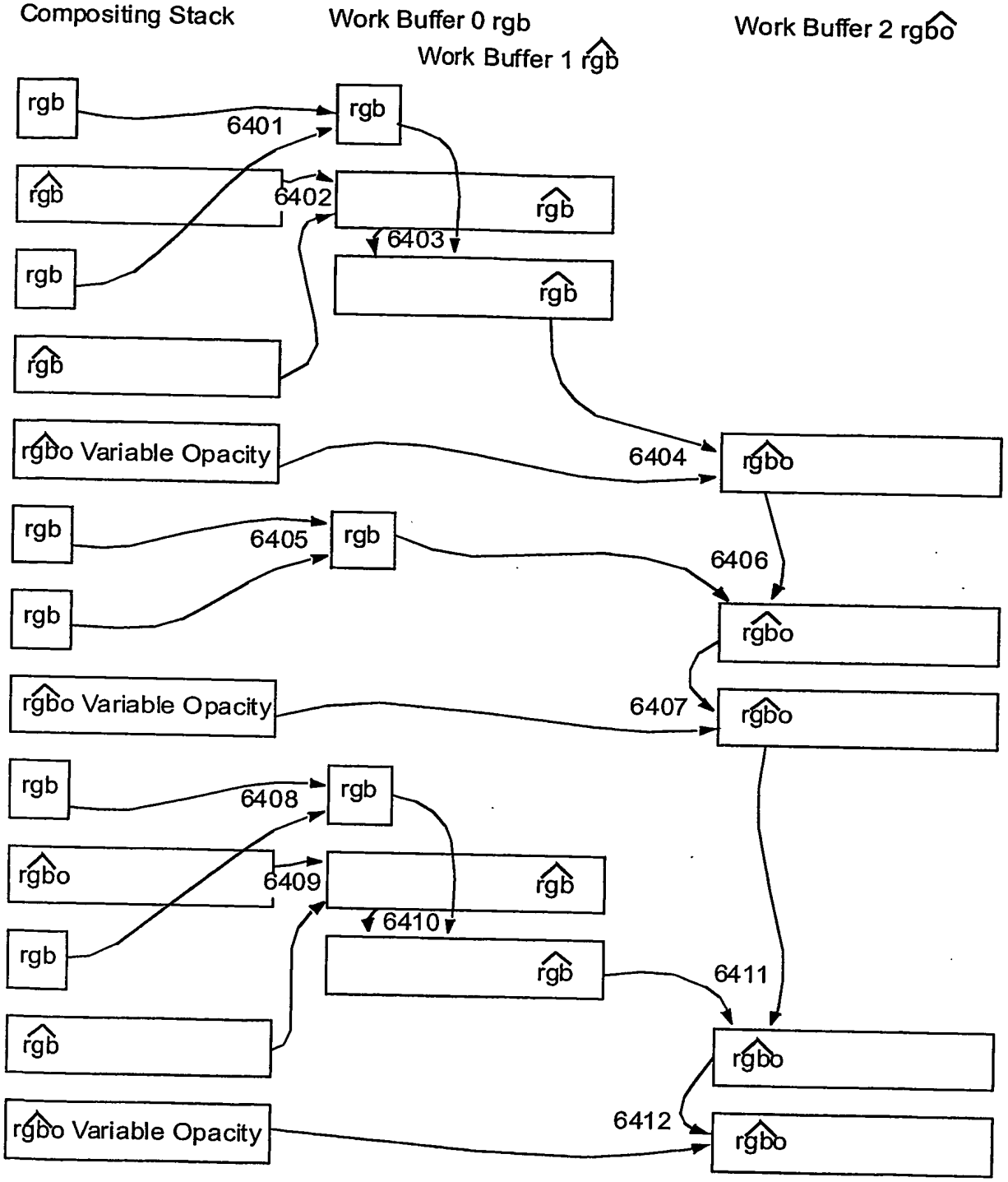

Compositing Stack
Work Buffer 0 rgb
Work Buffer 1 rgb
Work Buffer 2 rgbo
rgb
6401
6402
6403
rgbo Variable Opacity
6404
6405
6406
6407
6408
6409
6410
6411
6412

Fig. 64

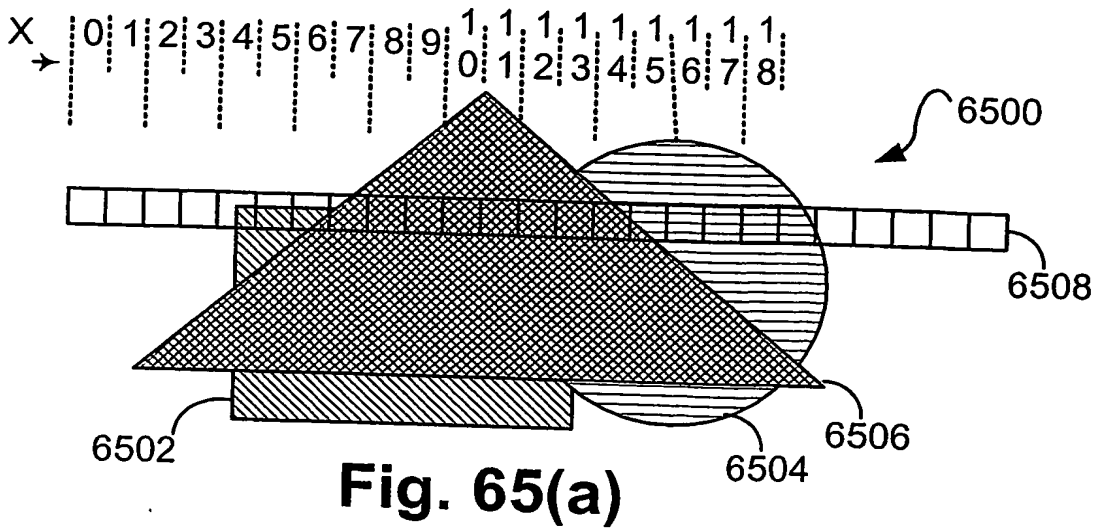


Fig. 65(a)

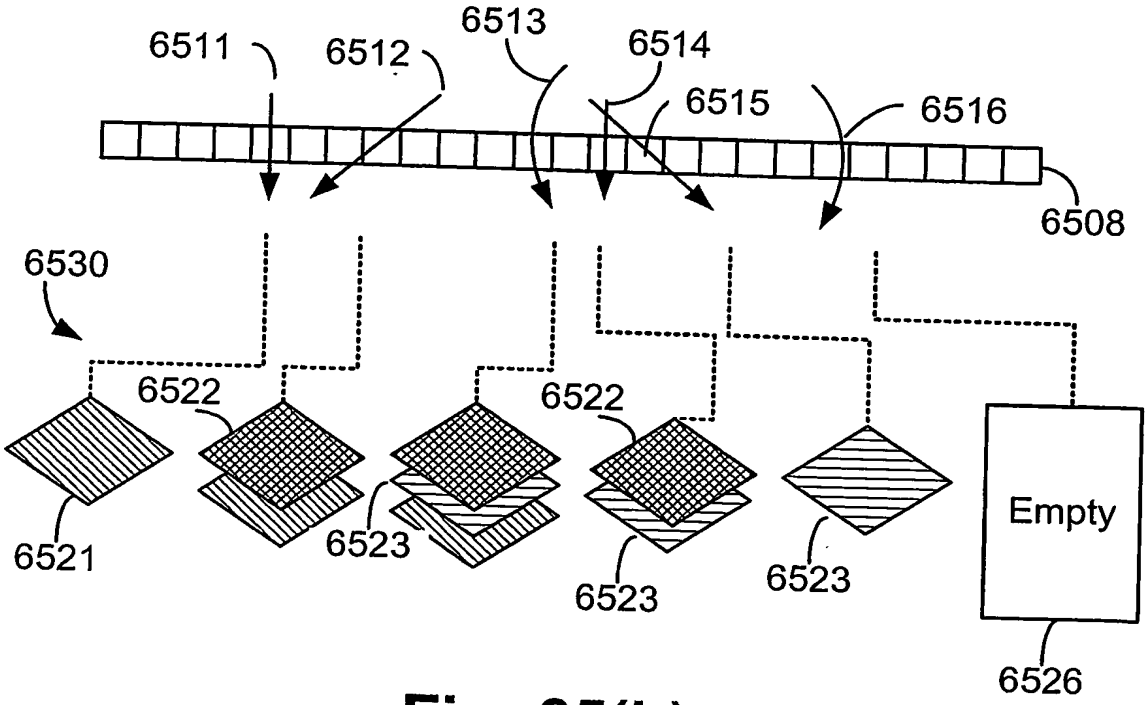


Fig. 65(b)

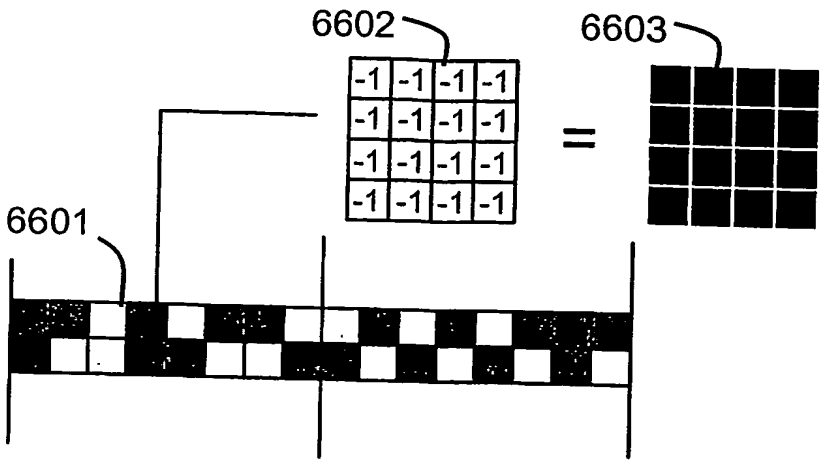


Fig. 66

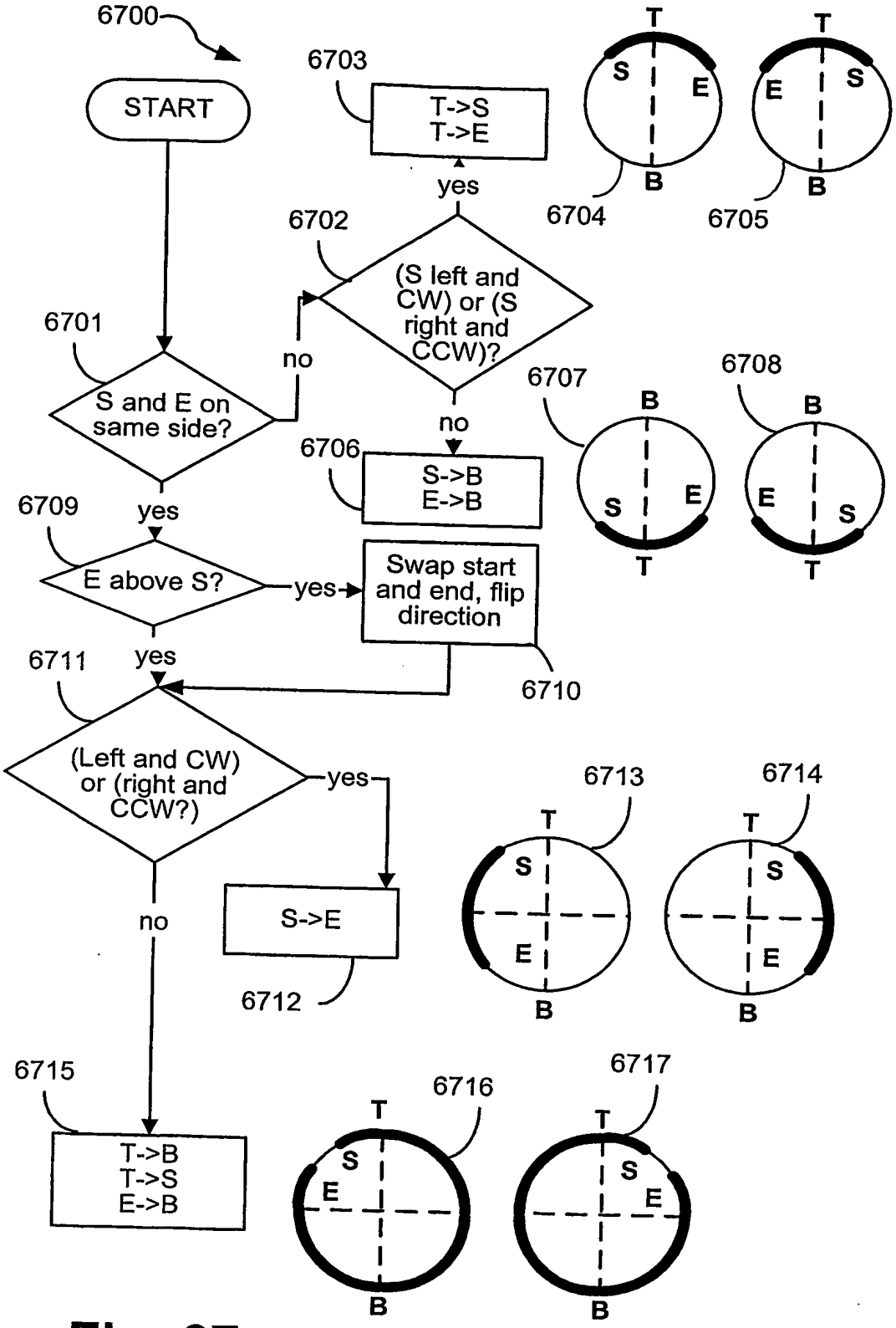

6700
START
6703
T->S
T->E
yes
6702
(S left and CW) or (S right and CCW)?
no
6701
S and E on same side?
no
6706
S->B
E->B
6709
yes
E above S?
yes
Swap start and end, flip direction
6710
6711
yes
(Left and CW) or (right and CCW?)
yes
S->E
6712
no
6715
T->B
T->S
E->B
6704
6705
6707
6708
6713
6714
6716
6717
T
S
E
B

Fig. 67

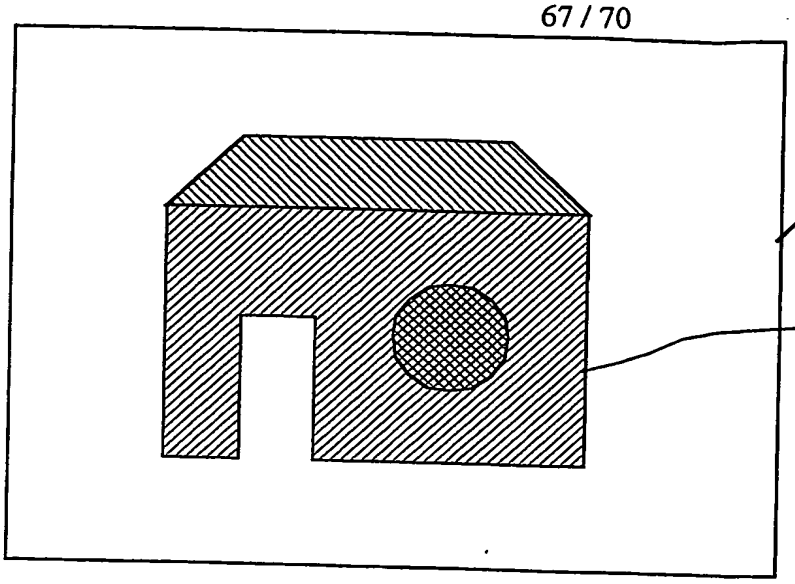


Fig. 68A

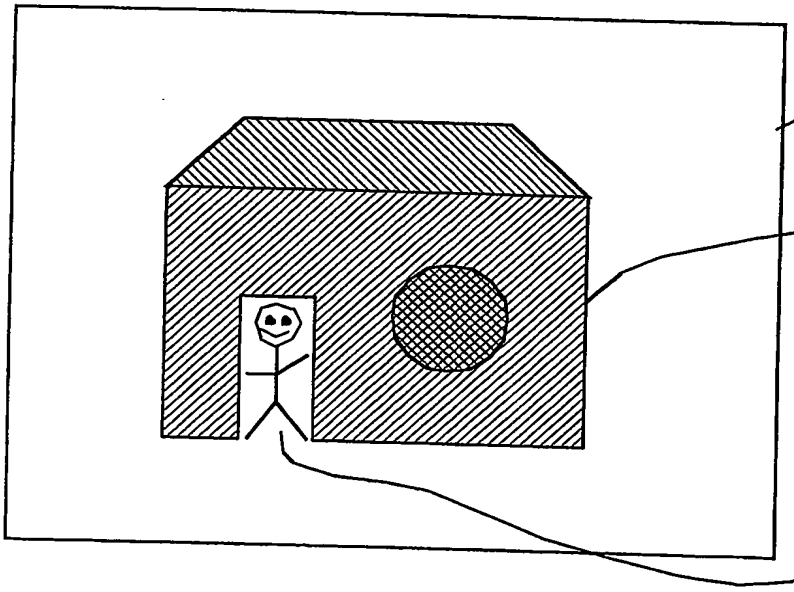


Fig. 68B

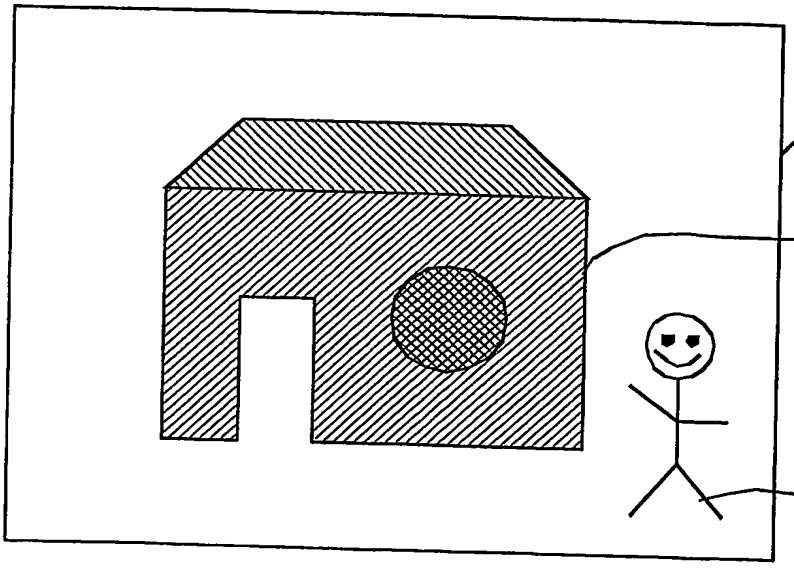


Fig. 68C

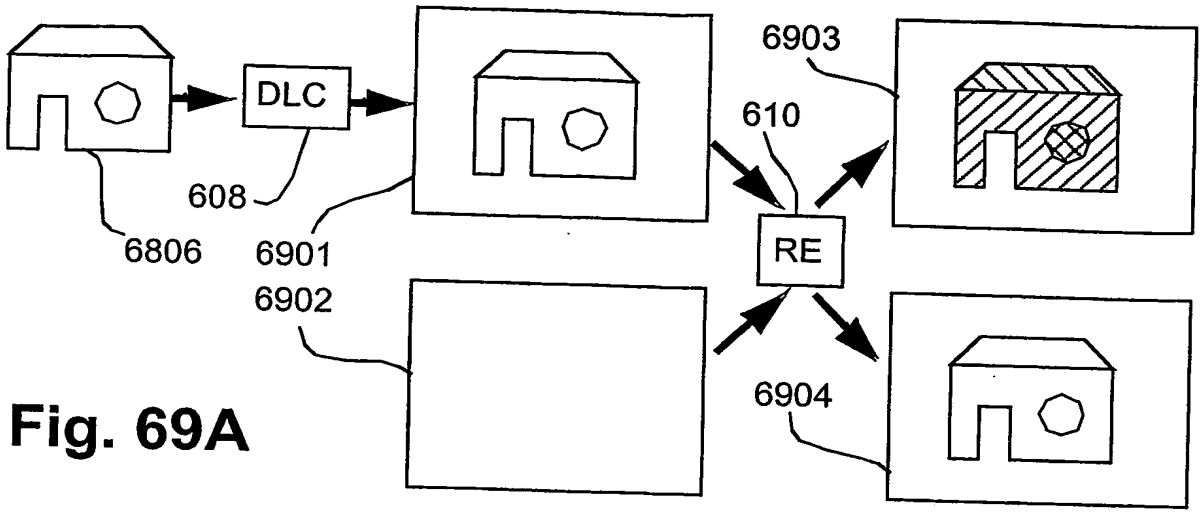


Fig. 69A

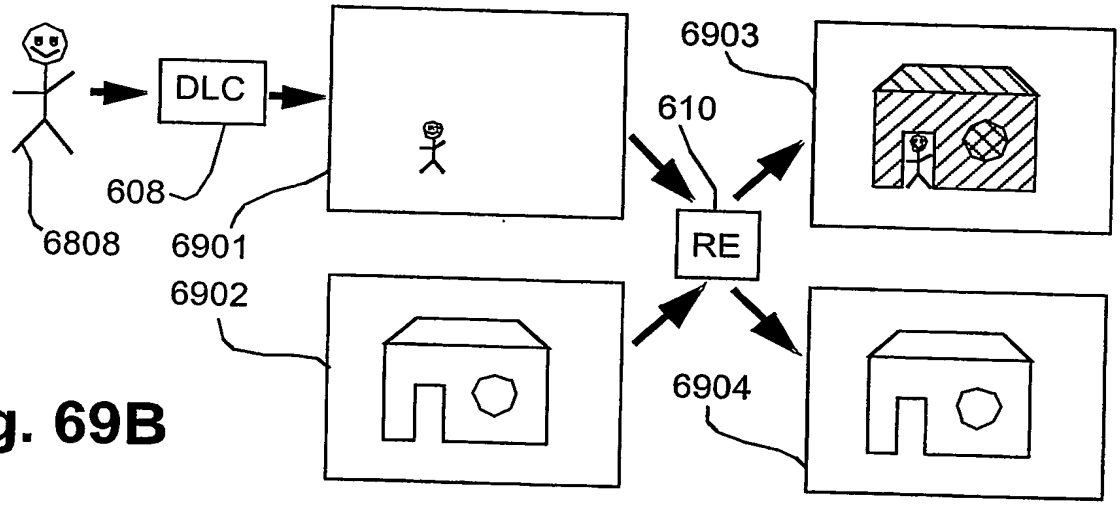


Fig. 69B

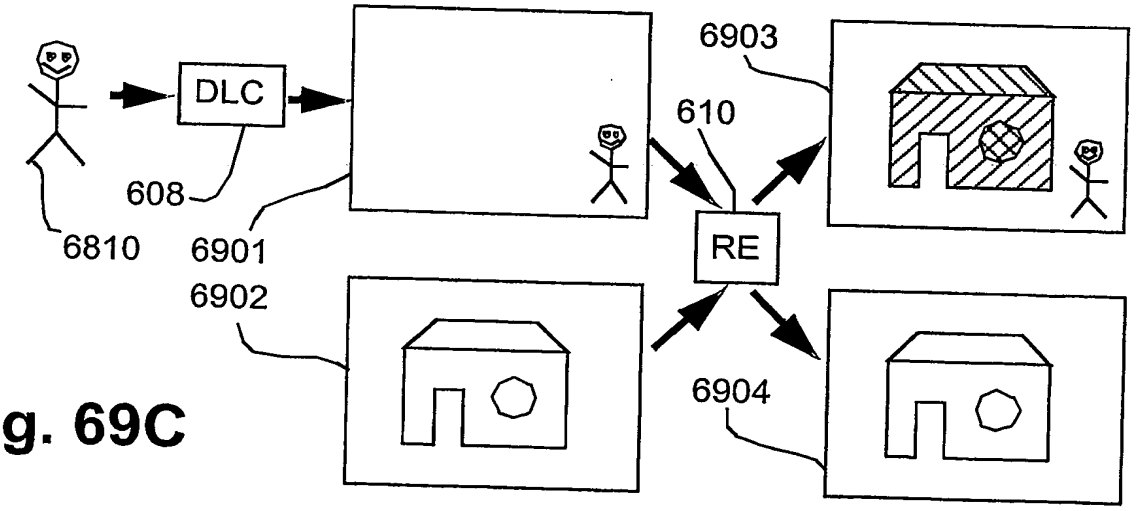


Fig. 69C

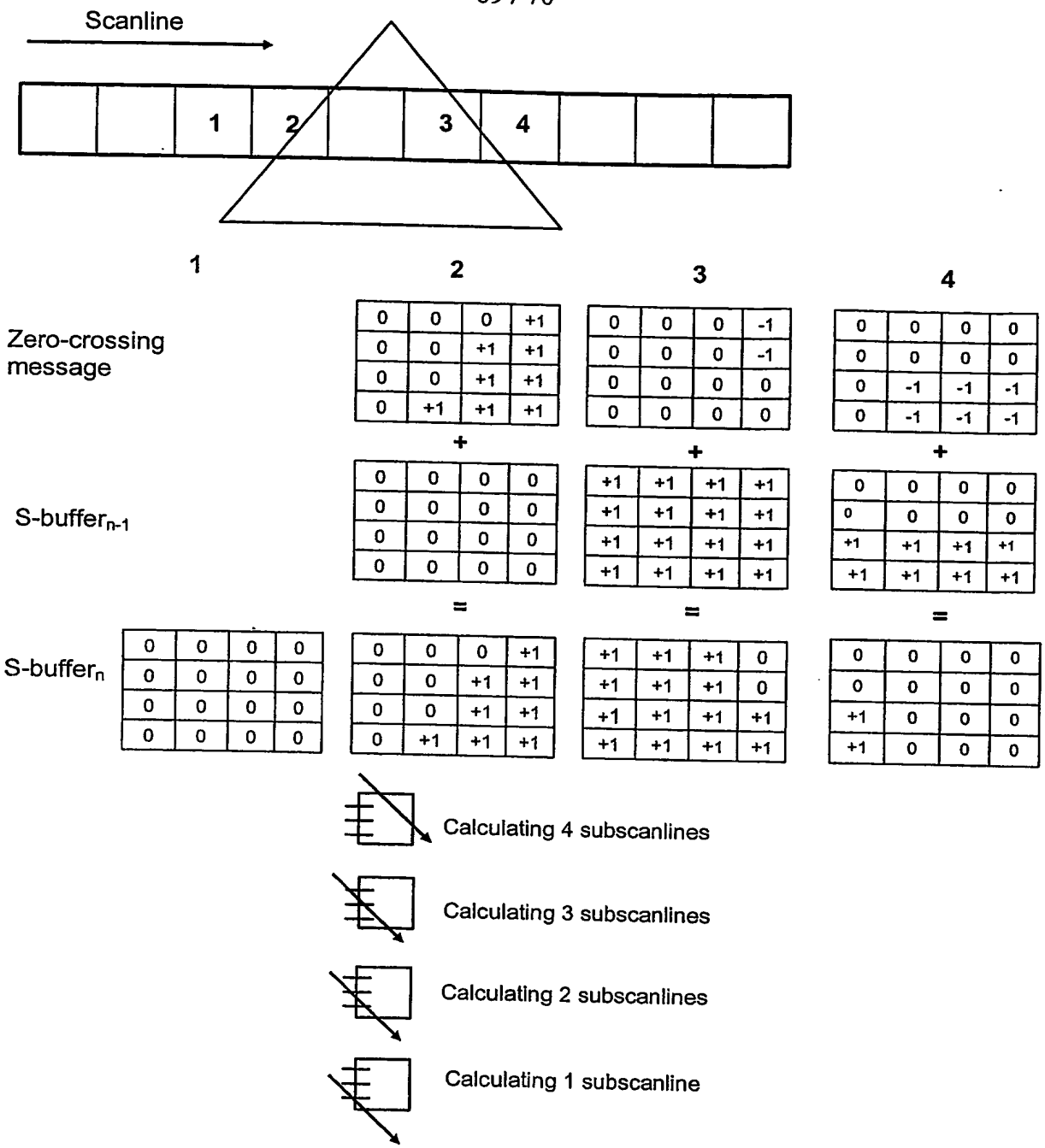

Scanline
1
2
3
4
1
2
3
4
Zero-crossing message
S-buffer$_{n-1}$
S-buffer$_n$
Calculating 4 subscanlines
Calculating 3 subscanlines
Calculating 2 subscanlines
Calculating 1 subscanline

Fig. 70

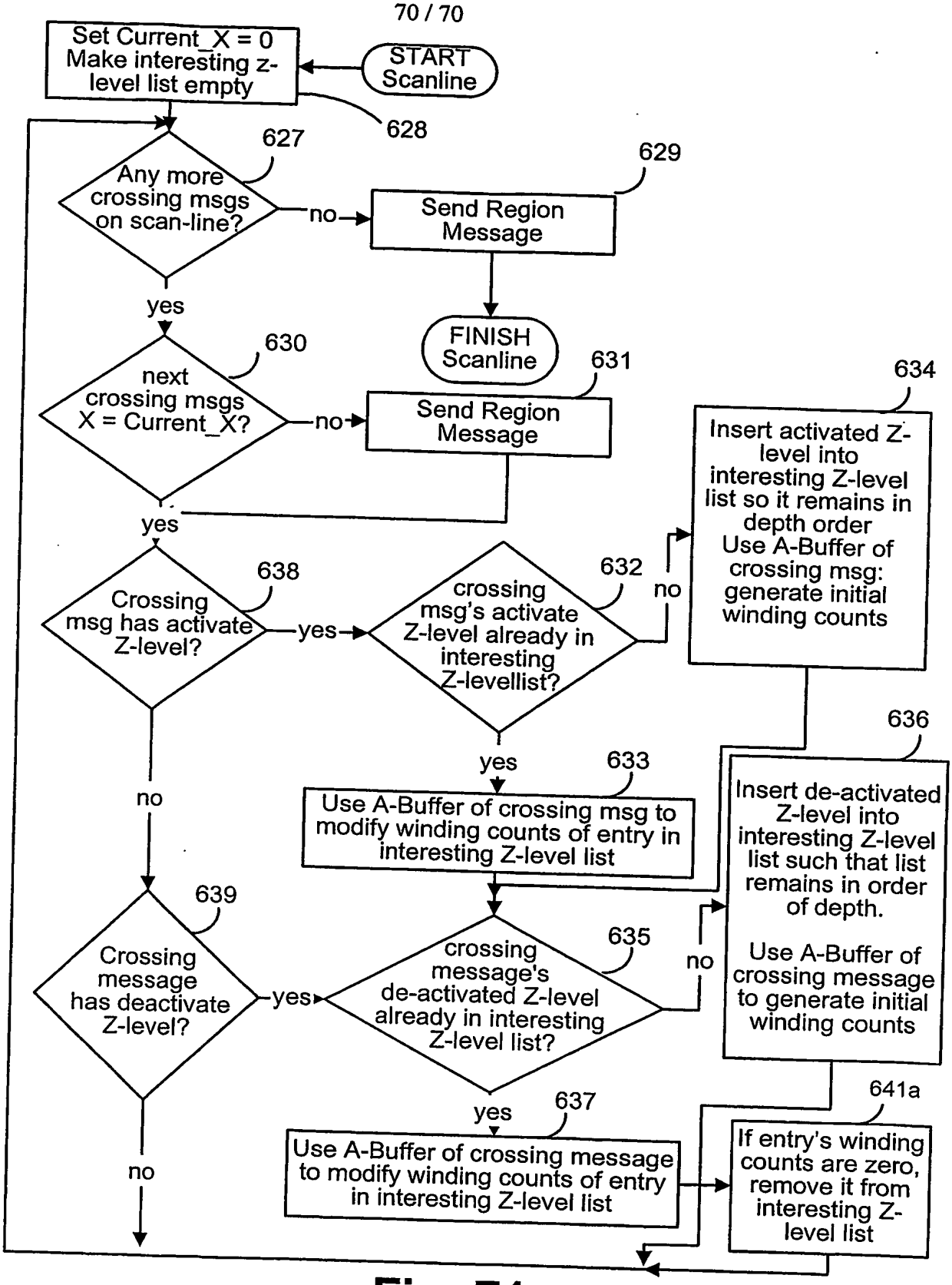

START Scanline
Set Current_X = 0 Make interesting z-level list empty
628
627
Any more crossing msgs on scan-line?
no
629
Send Region Message
FINISH Scanline
yes
630
next crossing msgs X = Current_X?
no
631
Send Region Message
634
Insert activated Z-level into interesting Z-level list so it remains in depth order Use A-Buffer of crossing msg: generate initial winding counts
yes
638
Crossing msg has activate Z-level?
yes
632
crossing msg's activate Z-level already in interesting Z-levellist?
no
636
Insert de-activated Z-level into interesting Z-level list such that list remains in order of depth.
Use A-Buffer of crossing message to generate initial winding counts
yes
633
Use A-Buffer of crossing msg to modify winding counts of entry in interesting Z-level list
no
639
Crossing message has deactivate Z-level?
yes
635
crossing message's de-activated Z-level already in interesting Z-level list?
no
641a
If entry's winding counts are zero, remove it from interesting Z-level list
yes
637
Use A-Buffer of crossing message to modify winding counts of entry in interesting Z-level list
no

Fig. 71